



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 145561

**PENGUKURAN METERING KWH PADA PELANGGAN
TEGANGAN RENDAH DENGAN MEDIA *POWER LINE
CARRIER***

Biner Rochmatul C.A
NRP 2214038002
M. Fatkhur Rozi
NRP 2214038021

Dosen Pembimbing
Ir. Josaphat Pramudijanto, M.Eng
Ir. Sjamsjul Anam, MT.

PROGRAM STUDI TEKNIK LISTRIK
Departemen Teknik Elektro Otomasi
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

-----*Halaman ini sengaja dikosongkan*-----



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - TE 145561

***MEASUREMENT METERING KWH AT LOW
VOLTAGE CUSTOMER WITH POWER LINE CARRIER***

Biner Rochmatul C.A
NRP 2214038002
M. Fatkhur Rozi
NRP 2214038021

Supervisor
Ir. Josaphat Pramudijanto, M.Eng
Ir. Sjamsjul Anam, MT.

ELECTRICAL ENGINEERING
Automation Electrical Engineering Department
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

-----*Halaman ini sengaja dikosongkan*-----

PERNYATAAN KEASLIAN
PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini kami menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir kami dengan judul **“Pengukuran Metering KWh Pada Pelanggan Tegangan Rendah Dengan Media Power Line Carrier”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.



Biner Rochmatul C.A
NRP 2214038002

Surabaya, 17 Juli 2017



M. Fatkhur Rozi
NRP 2214038021

-----*Halaman ini sengaja dikosongkan*-----

**PENGUKURAN METERING KWH PADA PELANGGAN
JARINGAN TEGANGAN RENDAH DENGAN MEDIA POWER
LINE CARRIER**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Memperoleh Gelar Ahli Madya Teknik
Pada

Program Studi Teknik Listrik
Departemen Teknik Elektro Otomasi
Fakultas Vokasi

Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I

Dosen Pembimbing II

Ir. Josaphat Pramudijanto, M.Eng.
NIP. 19621005 199003 1 003

Ir. Samsjul Anam, M.T.
NIP. 19630725 199003 1 002

**SURABAYA
JULI, 2017**

--- *Halaman ini sengaja dikosongkan* ---

PENGUKURAN METERING KWH PADA PELANGGAN TEGANGAN RENDAH DENGAN MEDIA *POWER LINE CARRIER*

Nama Mahasiswa	: Biner Rochmatul C.A
Nrp	: 2214 038 002
Nama Mahasiswa	: M. Fatkhur Rozi
Nrp	: 2214 038 021
Pembimbing 1	: Ir. Josaphat Pramudijanto. M.Eng.
NIP	: 19621005 199003 1 003
Pembimbing 2	: Ir. Sjamsjul Anam, M.T
NIP	: 19630725 199003 1 002

ABSTRAK

Peningkatan jumlah pelanggan energi listrik membuat PLN mengalami masalah dalam pencatatan dan juga pengukuran kWh oleh pelanggan energi listrik dikarenakan jumlah yang sangat banyak terutama pada pelanggan TR (Tegangan Rendah) dibawah 3,9 kVA dan juga dikarenakan jumlah yang sangat banyak membutuhkan waktu yang cukup lama untuk melakukan pencatatan.

Dalam tugas akhir ini akan dibuat sebuah alat pada Meter Trafo tiang dan juga alat pada meter pelanggan TR yang akan terhubung melalui media komunikasi PLC (*Power Line Carrier*) untuk memudahkan pencatatan KWh meter pelanggan dan juga mengurangi selisih pembacaan.

Pengukuran nilai KWh Meter digital dilakukan dengan mendeteksi *impuls* yang telah dihasilkan oleh KWh Meter Digital melalui sensor *Light Dependent Resistor* yang terdapat pada *Slave*. Tingkat kesalahan pengukuran KWh Meter yang digunakan yakni sebesar kurang lebih 1%. Jarak pengiriman data pengukuran kWh antara trafo tiang PHB dengan kWh meter pelanggan memiliki jarak uji sejauh 35 m. *Slave* melakukan pengiriman data pengukuran kWh kepada *Master* setiap pukul 10.00. Pengiriman data tersebut antar pelanggan memiliki *delay* sebanyak 2 detik untuk menghindari penumpukan data pengiriman.

Kata Kunci : PLC (*Power Line Carrier*), Pencatatan dan pengukuran , Jumlah pelanggan yang banyak.

-----*Halaman ini sengaja dikosongkan*-----

**MEASUREMENT METERING KWH AT LOW VOLTAGE
CUSTOMER WITH POWER LINE CARRIER**

<i>Student's Name</i>	: Biner Rochmatul C.A
<i>Registration Number</i>	: 2214 038 002
<i>Student's Name</i>	: M. Fatkhur Rozi
<i>Registration Number</i>	: 2214 038 021
<i>Supervisor 1</i>	: Ir. Josaphat Pramudijanto. M.Eng.
ID	: 19621005 199003 1 003
<i>Supervisor 2</i>	: Ir. Sjamsjul Anam, M.T
ID	: 19630725 199003 1 002

ABSTRACT

Increasing the number of electric energy customers makes PLN experience problems in recording and also the electrical power measurement by electric energy customers due to the very large number of customers especially TR (Low Voltage) below 3,9 kVA and also because the amount is very much takes a long time to do recording.

In this final project will be made a tool on Meter Transformer pole and also tool at meter TR customer that will be connected through PLC (Power Line Carrier) communication media to facilitate customer meter metering and also reduce the difference of reading.

Measurement of KWh Meter digital value is done by detecting impulses that have been produced by KWh Meter Digital through Light Dependent Resistor sensor contained in Slave. The measurement error rate of KWh Meter used is about 1%. The distance of the measurement data of kWh between the PHB pole transformer and the kWh meter customer has a test distance of 35 m. Slave sends the measurement data of kWh to Master every 10:00. Delivery of data between customers has a delay of 2 seconds to avoid the accumulation of data delivery.

Keywords : *Power Line Carrier (PLC), Defection and Size, Number of customers*

-----*Halaman ini sengaja dikosongkan*-----

KATA PENGANTAR

Puji Syukur kehadiran Allah SWT yang senantiasa memberikan ridho dan rahmat-Nya. Sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan tepat waktu.

Kami mengangkat Tugas Akhir dengan judul :
**“Pengukuran Metering KWh Pada Pelanggan Tegangan Rendah
Dengan Media *Power Line Carrier*”**

Dibuat guna memenuhi syarat kelulusan di Departemen Teknik Elektro Otomasi, bidang studi Teknik Listrik, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya.

Pada kesempatan ini penulis menyampaikan terima kasih atas segala doa, bantuan dan dukungannya yang telah diberikan selama proses pembuatan Tugas Akhir ini kepada :

1. Kedua orang tua atas dukungan baik spiritual maupun material yang tak ternilai harganya.
2. Bapak Ir. Joko Susila, MT. Ketua Departemen Teknik Elektro Otomasi, Fakultas Vokasi-ITS Surabaya.
3. Bapak Ir. Josaphat Pramudijanto, M Eng, sebagai Dosen Pembimbing yang telah banyak memberikan ilmu, pengarahan dan bimbingan selama penulis mengerjakan Tugas Akhir ini. Serta membimbing kami dengan kesabaran yang tiada batasnya.
4. Bapak Ir. Sjamsjul Anam , MT. sebagai Dosen pembimbing yang telah banyak membantu dalam kelancaran Tugas Akhir ini. Serta membimbing kami dengan kesabaran yang tiada batasnya.
5. Teman - teman di Departemen Teknik Elektro Otomasi ITS, yang telah banyak memberi bantuan dan semangat serta dukungan pada penulis.
6. Semua pihak yang telah membantu kami dalam penyelesaian Tugas Akhir ini yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari masih banyak kekurangan dalam tugas akhir ini. Kritik dan saran untuk pengembangan tugas akhir ini sangat diperlukan. Semoga tugas ini dapat bermanfaat bagi kita semua.

Surabaya, Juli 2017



Biner Rochmatul C.A
2214038002



M. Fatkhur Rozi
2214038021

DAFTAR ISI

HALAMAN

HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR... Error! Bookmark not defined.	
HALAMAN PENGESAHAN.....	vii
ABSTRAK.....	ix
ABSTRACT.....	x
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xviiiix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Permasalahan	2
1.3 Batasan Masalah	3
1.4 Tujuan.....	3
1.5 Sistematika Laporan.....	3
1.6 Relevansi.....	4
BAB II TEORI PENUNJANG.....	5
2.1 KWh Meter Elektronik.....	5
2.2 <i>Power Line Carrier</i> 6.....	
2.3 Arduino IDE.....	10
2.4 LCD (<i>Liquid Crystal Display</i>).....	11
2.5 <i>Light Dependent Resistor (LDR)</i>	13
2.6 <i>Push Button</i>	14
2.7 <i>Real Time Clock (RTC)</i>	15
2.8 Memori.....	16
2.9 <i>Board Arduino Mega</i>	17
2.10 <i>Serial Peripheral Interface (SPI)</i>	21
2.11 <i>Board Arduino Uno</i>	22
BAB III PERANCANGAN ALAT.....	25

3.1 Perancangan Sistem Keseluruhan.....	25
3.2 Perancangan <i>Hardware</i>	26
3.2.1 Perancangan <i>Prototype</i> Simulator	27
3.2.2 Perancangan KWh Meter	29
3.2.3 Perancangan <i>Slave</i>	30
3.2.3.1 Perancangan <i>Power Line Carrier</i> (PLC).....	31
3.2.3.2 Perancangan Arduino <i>Uno</i>	32
3.2.3.3 Perancangan LCD	32
3.2.3.4 Perancangan <i>Light Dependent Resistor</i> (LDR)	33
3.2.3.5 Perancangan LM 3805.....	34
3.2.3.6 Perancangan <i>Push Button</i>	35
3.2.4 Perancangan <i>Master</i>	35
3.2.4.1 Perancangan <i>Power Line Carrier</i> (PLC).....	36
3.2.4.2 Perancangan Arduino mega.....	37
3.2.4.3 Perancangan LCD.....	38
3.2.4.4 Perancangan <i>Light Dependent Resistor</i> (LDR).....	39
3.2.4.5 Perancangan LM 3805.....	40
3.2.4.6 Perancangan <i>Push Button</i>	40
3.2.4.7 Perancangan Memori Eksternal	41
3.2.4.8 Perancangan <i>Real Time Clock</i> (RTC).....	42
3.3 Perancangan <i>Software</i>	43
3.3.1 Perancangan <i>Software Slave</i>	43
3.3.2 Perancangan <i>Software Master</i>	46
 BAB IV PENGUJIAN DAN ANALISA DATA	 49
4.1 Pengujian <i>Slave</i>	49
4.1.1 Pengujian <i>Power Line Carrier</i> (PLC).....	52
4.1.2 Pengujian LCD	53
4.1.3 Pengujian <i>Light Dependent Resistor</i> (LDR).....	55
4.2 Pengujian <i>Master</i>	56
4.2.1 Pengujian <i>Power Line Carrier</i> (PLC).....	58
4.2.2 Pengujian LCD	59
4.2.3 Pengujian Memori Eksternal	60
4.2.4 Pengujian <i>Real Time Clock</i> (RTC).....	61
4.3 Pengujian KWh Meter.....	63

4.4 Pengujian <i>Protoype</i> Simulator.....	67
BAB V PENUTUP.....	71
5.1 Kesimpulan	71
5.2 Saran.....	71
DAFTAR PUSTAKA	73
LAMPIRAN A	A-1
A.1 Program <i>Slave</i>	A-1
A.2 Program LCD 2x16.....	A-12
A.3 Program <i>Light Dependent Resistor</i>	A-13
A.4 Program <i>Master</i>	A-14
A.5 Program LCD 20x4.....	A-16
A.6 Program Memori Eksternal	A-18
A.7 Program <i>Real Time Clock</i>	A-20
LAMPIRAN B	B-1
B.1 <i>Datasheet Real Time Clock</i>	B-1
B.2 <i>Datasheet</i> LM 3805.....	B-15
B.3 <i>Datasheet</i> Arduino Uno.....	B-27
B.4 <i>Datasheet</i> Arduino Mega 2560.....	B-31
RIWAYAT HIDUP PENULIS	C-1

-----*Halaman ini sengaja dikosongkan*-----

DAFTAR GAMBAR

HALAMAN

Gambar 2.1	KWh Meter Elektronik	6
Gambar 2.2	Blok Diagram <i>Power Line Carrier (PLC)</i>	9
Gambar 2.3	Jendela Arduino IDE.....	10
Gambar 2.4	Skema Rangkaian LCD 20 X 4.....	12
Gambar 2.5	Skema Rangkaian <i>Light Dependent Resistor</i>	14
Gambar 2.6	Konstruksi <i>Light Dependent Resistor</i>	14
Gambar 2.7	Skema Rangkaian <i>Push Button</i>	15
Gambar 2.8	Skematik RTC DS3231	16
Gambar 2.9	<i>Memory Card</i>	17
Gambar 2.10	<i>Board</i> Arduino Mega.....	17
Gambar 2.11	Blok Diagram SPI	22
Gambar 2.12	<i>Board</i> Arduino Uno.....	23
Gambar 3.1	Rancangan Sistem Alat	26
Gambar 3.2	Diagram Fungsional Alat Keseluruhan	26
Gambar 3.3	Perancangan <i>Prototype</i> Simulator	27
Gambar 3.4	<i>Wiring</i> KWh Meter.....	30
Gambar 3.5	Perancangan <i>Slave</i>	30
Gambar 3.6	Perancangan <i>Power Line Carrier</i> dengan Arduino Uno	31
Gambar 3.7	Perancangan Arduino Uno	32
Gambar 3.8	Perancangan <i>LCD</i> dengan Arduino Uno	33
Gambar 3.9	Perancangan <i>Light Dependent Resistor</i> dengan Arduino Uno	34
Gambar 3.10	Perancangan <i>Push Button</i> dengan Arduino Uno	35
Gambar 3.11	Perancangan <i>Master</i>	36
Gambar 3.12	Perancangan <i>Power Line Carrier</i> dengan Arduino Mega.....	37
Gambar 3.13	Perancangan Arduino Mega	38
Gambar 3.14	Perancangan <i>LCD</i> dengan Arduino Mega	39
Gambar 3.15	Perancangan <i>Light Dependent Resistor</i> dengan Arduino Mega	40
Gambar 3.16	Perancangan <i>Push Button</i> dengan Arduino Mega	41
Gambar 3.17	<i>Wiring</i> Modul <i>Micro SD Adapter</i> dengan Arduino Mega	42
Gambar 3.18	<i>Wiring</i> Modul RTC DS1307 dengan Arduino Mega.....	43
Gambar 3.19	<i>Flowchart</i> Perancangan <i>Software Slave</i>	45

Gambar 3.20	Perancangan Pemrograman <i>Slave</i>	46
Gambar 3.21	<i>Flowchart</i> Perancangan <i>Software Master</i>	47
Gambar 3.22	Perancangan Pemrograman <i>Master</i>	48
Gambar 4.1	Pengujian <i>Slave</i>	51
Gambar 4.2	Pengujian <i>Power Line Carrier</i>	53
Gambar 4.3	Pengujian LCD	54
Gambar 4.4	Pengujian LDR.....	56
Gambar 4.5	Pengujian <i>Master</i>	57
Gambar 4.6	Pengujian LCD 20 X 4.....	60
Gambar 4.7	Pengujian Memori Eksternal	61
Gambar 4.8	Pengujian <i>Real Time Clock</i> (RTC).....	62
Gambar 4.9	Pengujian Tingkat Kesalahan KWh Meter.....	63
Gambar 4.10	Tampilan Alat Uji KWh Meter	64
Gambar 4.11	Pengujian <i>Prototype</i> Simulator	68

DAFTAR TABEL

HALAMAN

Tabel 2.1	Spesifikasi <i>Power Line Carrier</i>	9
Tabel 2.2	Konfigurasi Pin LCD 20x4	12
Tabel 4.1	Pengujian <i>Impuls</i>	50
Tabel 4.2	Pengujian <i>Slave</i>	51
Tabel 4.3	Pengujian Jarak <i>Power Line Carrier</i>	53
Tabel 4.4	Pengujian <i>Light Dependent Resistor</i> (LDR)	55
Tabel 4.5	Pengujian <i>Master</i>	57
Tabel 4.6	Pengujian Jarak <i>Power Line Carrier</i>	59
Tabel 4.7	Pengujian Memori Eksternal	61
Tabel 4.8	Pengujian Pengujian <i>Real Time Clock</i> (RTC)	62
Tabel 4.9	Pengujian Tingkat Kesalahan KWh Meter <i>Master</i>	64
Tabel 4.10	Pengujian Tingkat Kesalahan KWh Meter Pelanggan A ...	65
Tabel 4.11	Pengujian Tingkat Kesalahan KWh Meter Pelanggan B....	65
Tabel 4.12	Pengujian Perhitungan KWh Meter Pelanggan A	66
Tabel 4.13	Pengujian Perhitungan KWh Meter Pelanggan B	66
Tabel 4.14	Pengujian <i>Prototype</i> Simulator	69
Tabel 4.15	Pengujian Waktu <i>Prototype</i> Simulator	69

-----*Halaman ini sengaja dikosongkan*-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan terhadap energi belakangan ini semakin meningkat terutama penggunaan tenaga listrik. Hampir sebagian kegiatan masyarakat di Indonesia sangat mengandalkan listrik sebagai penyokong kehidupan mereka. Seiringnya peningkatan penggunaan energi listrik, PLN sebagai satu-satunya perusahaan pendistribusi energi listrik diharapkan agar dapat memenuhi kebutuhan energi listrik tersebut. Namun yang sangat disayangkan dengan peningkatan jumlah pelanggan energi listrik tersebut, PLN mengalami masalah dalam pencatatan dan juga pengukuran kWh pelanggan dikarenakan jumlah pelanggan yang sangat banyak terutama pada pelanggan TR (Tegangan Rendah) yang berlangganan dibawah 3,9 kVA. Umumnya pada pelanggan dibawah 3,9 kVA ini pencatatan yang digunakan adalah secara manual. Yakni para petugas pencatat meter (Cater) harus mendatangi rumah pelanggan secara langsung untuk membaca besar kWh yang digunakan oleh pelanggan tiap bulannya. Hal ini berbeda dengan pelanggan yang berlangganan diatas 3,9 kVA karena pada pelanggan ini menggunakan AMR (*Automatic Meter Reading*) sebagai aplikasi dalam pembacaan KWh meter. Pembacaan penggunaan kWh oleh para Cater terkadang terdapat beberapa kesalahan, seperti halnya pada KWh meter yang mengalami kerusakan disisi KWh meternya maupun dari sisi sikap kooperatif para pelanggan. Pembacaan Cater yang memiliki jangka pembacaan selama 10 hari juga mempengaruhi selisih dari ketelitian pembacaan Cater tiap pelanggannya. Saat ini PLN sendiri telah mengembangkan penggunaan KWh meter pada panel PHB guna memudahkan perbandingan data dari tiap daerah yang disuplai oleh panel PHB tersebut sehingga dapat mengurangi selisih dari ketidak telitian pembacaan pada meter pelanggan. Selain pengembangan dalam pengukuran kWh pada panel PHB, PLN sedang menggalakkan gerakan penggantian KWh meter mekanik (piringan) dengan KWh meter digital. Penggantian jenis KWh meter ini juga salah satu cara menanggulangi kecurangan yang dilakukan pelanggan dengan memodifikasi dari KWh meter mekanik.

Ada beberapa alternatif yang mungkin dikembangkan untuk menanggulangi masalah tersebut, salah satu diantaranya adalah aplikasi teknologi *Wifi* sebagai pengiriman data layaknya penggunaan *Wifi* pada

AMR pada pelanggan diatas 3,9 kVA. Namun aplikasi ini juga sangat sulit diterapkan, mengingat lokasi daerah pedalaman yang umumnya sangat jauh dan terhalang oleh bukit dan gunung, sehingga diperlukan tambahan *repeater* pada daerah-daerah tertentu. Harga yang mahal juga menghalangi penggunaan aplikasi ini pada pelanggan dengan penggunaan daya dibawah 3,9 kVA. Salah satu alternatif lain yang mungkin untuk dikembangkan adalah *Power Line Carrier* (PLC). *Power Line Carrier* (PLC) berfungsi sebagai media penyalur data pengukuran KWh Meter para pelanggan kepada KWh Meter Trafo Tiang. Alternatif ini merupakan salah satu bentuk teknologi informasi melalui jaringan listrik. Hal ini tentu saja sangat dimungkinkan oleh karena energi listrik merupakan komponen utama dalam menjalankan perangkat elektronik dan peralatan komunikasi lainnya. Sehingga dapat dikatakan bahwa infrastruktur IT hanya bisa dibangun pada suatu daerah jika pada daerah itu sudah ada jaringan listrik. Di sisi lain, PLN sebagai BUMN yang bertanggung jawab dalam penyediaan tenaga listrik beberapa waktu yang lalu telah mencanangkan program 75-100 (pada usia kemerdekaan RI yang ke-75 tahun, PLN menjamin rasio elektrifikasi 100 %) hal ini berarti PLN sedang menuju ke pelayanan penyediaan listrik ke seluruh pelosok tanah air. Penggantian KWh meter mekanik menjadi KWh meter digital juga menjadi aspek penting guna mendukung alternatif teknologi yang ditawarkan hingga saat ini. Data pengukuran kWh pada KWh meter digital dapat dilakukan dengan pengukuran nilai *implus* yang dihasilkan oleh KWh meter. Sehingga sangat dimungkinkan aplikasi PLC dapat memenuhi kebutuhan informasi seluruh masyarakat di tanah air tanpa terkecuali.

Oleh karena itu, kami berinisiatif untuk membuat Tugas Akhir yang berjudul “Pengukuran Matering KWh Pada Pelanggan Tegangan Rendah Dengan Media *Power Line Carrier*”. Yakni pembuatan sebuah pengukuran KWh Meter pada panel PHB dan juga KWh meter pada pelanggan TR yang akan terhubung melalui media komunikasi PLC (*Power Line Carrier*) untuk memudahkan pembacaan KWh meter pelanggan dan juga mengurangi selisih pembacaan.

1.2 Permasalahan

Pada Tugas Akhir ini yang menjadi permasalahan utama yaitu permasalahan PLN dalam hal pencatatan dan juga pengukuran KWh meter listrik oleh pelanggan energi dikarenakan jumlahnya yang sangat banyak terutama pada pelanggan TR (Tegangan Rendah) satu fasa yang

berlangganan dibawah 3,9 kVA dan juga dikarenakan jumlahnya yang sangat banyak membutuhkan maka waktu yang cukup lama untuk melakukan pencatatan. Selain itu juga disebabkan oleh jumlah SDM untuk melakukan pencatatan dari tiap pelanggan waktunya berbeda sehingga lama waktu pemakaian dari tiap pelanggan berbeda. Hal itu menyebabkan adanya selisih nilai pembacaan antara penggunaan kWh yang satu dengan yang lain.

1.3 Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari Tugas Akhir ini adalah :

1. Pencatatan dan pengukuran nilai kWh pada KWh Meter pasca bayar pelanggan listrik Tegangan Rendah Satu Fasa yang berlangganan dibawah 3,9 kVA.
2. Penyaluran data pencatatan KWh meter listrik pelanggan tegangan rendah dengan menggunakan media *Power Line Carrier*.

1.4 Tujuan

Tujuan Biner Rochmatul C. A menuliskan Tugas Akhir ini adalah:

1. Merancang dan membuat pengukuran nilai kWh pada KWh Meter pelanggan.
2. Merancang dan membuat proses pengiriman data pada KWh Meter pelanggan melalui media *Power Line Carrier* (PLC).

Dan tujuan M. Fatkhur Rozi menuliskan Tugas Akhir ini adalah:

1. Merancang dan membuat pengukuran nilai kWh pada KWh Meter trafo tiang.
2. Merancang dan membuat proses penerimaan data pengukuran nilai kWh dari KWh Meter pelanggan melalui media *Power Line Carrier* (PLC).

1.5 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I :PENDAHULUAN

Dalam bab ini menjelaskan tentang latar belakang, permasalahan, tujuan penelitian, sistematika laporan, dan relevansi.

Bab II :TEORI PENUNJANG

Dalam bab ini menjelaskan tentang tinjauan pustaka, konsep dari KWh meter digital, Arduino uno, Arduino Mega, *Serial Peripheral Interface* (SPI), sensor *Light Dependent Resistor* (LDR), *Real Time Clock* (RTC), Memori, LCD, dan sistem komunikasi data menggunakan *Power Line Carrier*.

Bab III :PERANCANGAN ALAT

Dalam bab ini membahas perancangan sistem *hardware* maupun *software* pada Pengukuran Metering KWh Pada Pelanggan Tegangan Rendah Dengan Media *Power Line Carrier* berdasarkan teori dasar pada Bab II.

Bab IV :PENGUJIAN DAN ANALISA DATA

Dalam bab ini membahas tentang pengukuran, pengujian, serta analisa terhadap prinsip kerja dan proses dari suatu alat yang dibuat.

Bab V :PENUTUP

Dalam bab ini membahas berisi tentang penutup yang menjelaskan tentang kesimpulan dari Tugas Akhir dan saran – saran untuk pengembangan alat ini lebih lanjut.

1.6 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan dengan pengembangan lebih lanjut dapat dijadikan sebagai referensi atau ide dalam kemajuan sistem metering pada Distribusi tenaga listrik, khususnya sistem metering kWh pada pelanggan tegangan rendah. Serta dapat memberikan inovasi kepada PT PLN (Persero) agar mempermudah dalam pekerjaan pencatatan meter.

BAB II

TEORI PENUNJANG

Pada bab ini akan dibahas dasar teori secara keseluruhan yang digunakan oleh M. Fatkhur Rozi dan Biner Rochmatul C.A. Dasar teori ini mencakup teori penunjang serta peralatan yang digunakan untuk membuat rancangan dari sistem pengukuran metering kWh pada pelanggan tegangan rendah dengan media *Power Line Carrier*. M. Fatkhur Rozi dan Biner Rochmatul C.A membutuhkan teori penunjang dari KWh meter elektronik, *Light Dependent Resistor (LDR)*, *Push Button*, *Power Line Carrier*, LCD, dan arduino IDE. Teori penunjang tersebut terdapat pada Sub Bab 2.1 KWh Meter Elektronik, 2.2 *Power Line Carrier (PLC)*, 2.3 Arduino IDE, 2.4 LCD, 2.5 *Light Dependent Resistor (LDR)*, dan 2.6 *Push Button*.

Dan M. Fatkhur Rozi membutuhkan teori penunjang dari Arduino Mega, *Real Time Clock (RTC)*, serta Memori. Teori penunjang tersebut terdapat pada Sub Bab 2.7 *Real Time Clock (RTC)*, 2.8 Memori, 2.9 Arduino Mega, dan 2.10 *Serial Peripheral Interface (SPI)*.

Juga Biner Rochmatul C.A membutuhkan teori penunjang dari Arduino Uno. Teori penunjang tersebut terdapat pada Sub Bab 2.11 Arduino Uno.

M. Fatkhur Rozi dan Biner Rochmatul C.A membutuhkan beberapa teori penunjang diantaranya yakni :

2.1 KWh Meter Elektronik [1]

KWh meter adalah alat pengukur energi listrik yang mengukur secara langsung hasil kali tegangan, arus factor kerja, kali waktu yang tertentu ($UI \cos \phi t$) yang bekerja padanya selama jangka waktu tertentu tersebut. Hal ini berdasarkan bekerjanya induksi magnetis oleh medan magnet yang dibangkitkan oleh arus melalui kumparan arus terhadap *disc* (piring putar) KWh meter, dimana induksi magnetis ini berpotongan dengan induksi magnetis yang dibangkitkan oleh arus melewati kumparan tegangan terhadap *disc* yang sama.

Koppel putar dapat dibangkitkan terhadap *disc* karena induksi magnetis kedua medan magnet tersebut diatas bergeser fasa sebesar 90° satu terhadap lainnya (azas Ferrari). Hal ini dimungkinkan dengan konstruksi kumparan tegangan dibuat dalam jumlah besar gulungan sehingga dapat dianggap induktansi murni.

Meter elektronik bekerja berdasarkan prinsip *elektronis*. Sinyal arus dan tegangan diteruskan ke sinyal *processor modul*, meliputi modul modul : *Transformer Modul*, *Power Supply Modul*, *Analog To Digital Modul*, *Register Processor Modul*, *Display Modul*, *Mass Memory Modul*, *Input / Output Modul*, dan *Communication Modul*. Sedangkan meter elektro mekanik bekerja berdasarkan prinsip elektro mekanik. Arus dan tegangan listrik menimbulkan gaya gerak listrik yang menggerakkan / memutar piringan pada porosnya. Putaran poros piringan diteruskan melalui roda-roda gigi ke *drum register*. Konstruksi meter elektronik dapat dilihat pada Gambar 2.1 KWh Meter Elektronik.



Gambar 2.1 KWh Meter Elektronik

2.2 *Power Line Carrier* [2][3]

Power Line Carrier (PLC) merupakan suatu perangkat komunikasi yang memanfaatkan konduktor jaringan listrik sebagai media komunikasi, yaitu berupa penyaluran data dengan frekuensi tinggi yang ditumpangkan pada gelombang listrik, sehingga bisa dikatakan bahwa selain sebagai penyalur tenaga listrik, konduktor atau kabel listrik tersebut mampu digunakan sebagai penyalur data pula.

Dalam penggunaannya, ada beberapa hal yang dapat dilakukan dalam penggunaan PLC. Misalnya PLC dapat digunakan untuk akses internet melalui tegangan listrik dari PLN karena tegangan yang masuk ke rumah mengandung data dan *voice*, sehingga dapat digunakan untuk internet tidak diperlukan kabel khusus. Teknologi ini jauh lebih efisien dibandingkan dengan penggunaan kabel pada umumnya, karena dengan teknologi ini pemilik rumah dapat mengakses internet dan menonton acara televisi pada saat yang bersamaan tanpa harus menggunakan kabel

tambahan yang panjang. Penggunaan PLC dapat dilihat pada Gambar 2.2 Blok Diagram *Power Line Carrier*.

Dalam penggunaan teknologi PLC ini kita dapat memilih sebagai salah satu solusi teknologi jaringan akses yang sangat menguntungkan. Namun dalam implementasinya dilapangan perlu diperhatikan masalah *noise* atau gangguan yang terjadi. Dalam pengiriman dan penerimaan data sinyal, kita harus benar-benar menyamakan atau mensinkronkan antara PLC yang satu dengan yang lain agar data tersebut dapat diterima sesuai dengan yang kita harapkan.

Bagian utama sistem komunikasi menggunakan PLC adalah sebagai berikut :

1. Bagian Pemancar (*Transmitter*).
Sinyal informasi berupa sinyal analog maupun digital ditumpangkan ke sinyal pembawa melalui teknik modulasi sehingga dihasilkan sinyal pembawa termodulasi (sinyal lolos pita), kemudian diperkuat untuk dipancarkan melalui media transmisi saluran distribusi daya.
2. Rangkaian Gandengan (*Line Coupling*)
Rangkaian gandengan terdiri atas kapasitor gandengan yang berfungsi untuk mengisolasi peralatan komunikasi dari tegangan jala-jala listrik. Fungsi ini dipenuhi dengan memberikan impedansi rendah ke frekuensi pembawa dan memberikan impedansi tinggi pada frekuensi jala-jala listrik. Rangkaian yang kedua berupa penala jalur yang berfungsi mengkompensasi reaktan kapasitif dan sebagai penyesuaian impedansi antara saluran daya atau jala-jala listrik dengan peralatan komunikasi.
3. Media Transmisi
Berupa kabel saluran transmisi daya atau jala-jala listrik yang digunakan sebagai antena pada sistem komunikasi melalui saluran distribusi daya.
4. Bagian Penerima (*Receiver*)
Melakukan penguraian atau pendemodulasian sinyal pembawa termodulasi yang diterima dimana teknik yang digunakan sama dengan di pemancar serta melakukan sinkronisasi antara pemancar dan penerima dengan jalan pemulihan sinyal pembawa yang diterima sehingga diperoleh kembali sinyal informasi yang dikirimkan. Dalam distribusi daya yang perlu diperhatikan adalah bagaimana menyalurkan

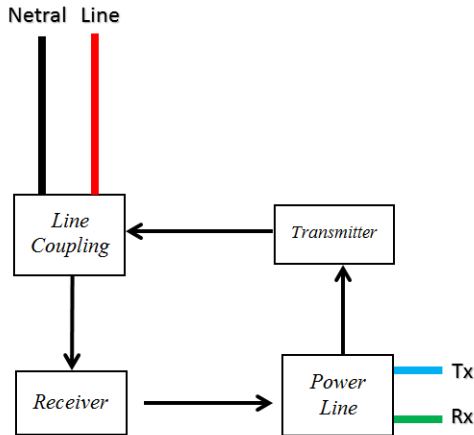
energi listrik dengan rugi-rugi sekecil mungkin, sedangkan dalam sistem komunikasi yang perlu diperhatikan adalah bagaimana informasi yang dikirim dapat diterima dengan kualitas yang baik.

Pembawa pada saluran daya (PLC) adalah salah satu telekomunikasi yang sinyal pembawa-nya ditumpangkan (*superposed*) pada jaringan tenaga listrik. Frekuensi sinyal / arus pembawa berbeda dengan frekuensi arus jaringan tenaga listrik, sehingga menjadikan dia jaringan rangkaian transmisi berfrekuensi tinggi. Dalam hal ini dikenal dua cara hubungan komunikasi dengan jaringan tenaga listrik, yakni:

1. PLC (*Power Line Carrier*) terhubung induktif, yaitu dengan menempatkan penghantar jaringan listrik untuk jarak tertentu.
2. PLC (*Power Line Carrier*) terhubung kapasitif, yaitu menghubungkan peralatan komunikasi dengan jaringan tenaga listrik lewat kapasitor.

Power Line Carrier yang dibutuhkan sebagai media komunikasi adalah PLC yang ketika tidak sedang digunakan untuk mengirimkan data tidak memunculkan sinyal *carrier*, dikarenakan munculnya sinyal *carrier* yang muncul ketikad tidak adanya aktifitas pengiriman data bisa menyebabkan terjadinya kesalahan pembacaan dari PLC penerima ketika terjadinya pengiriman.

Dengan jumlah *Slave* yang lebih banyak dibandingkan dengan *Master* perlu di buat suatu *protocol* yang mana dapat mencegah terjadinya kesalahan pada pengiriman dikarenakan data yang menumpuk. Salah satu contohnya adalah dengan cara membuat waktu pengiriman data dari tiap *Slave* ke *Master* bergantian sesuai dengan permintaan dari *Master*.



Gambar 2.2 Blok Diagram *Power Line Carrier (PLC)*

PLC (*Power Line Carrier*) yang digunakan di sini memanfaatkan jaringan listrik tegangan rendah (220 Volt). *Power Line Carrier* yang digunakan memiliki beberapa spesifikasi khusus yakni pada Tabel 2.1 Spesifikasi *Power Line Carrier*. Berikut merupakan spesifikasi dari *Power Line Carrier*:

Tabel 2.1 Spesifikasi *Power Line Carrier*

No	Spesifikasi <i>Power Line Carrier</i>	Keterangan
1	Tegangan	220V/110V AC Atau 12V - 16V DC
2	<i>Baud Rate</i> TTL UART	1200 2400 4800 9600 (<i>default 9600</i>) <i>piority check, no parity optional</i>
3	<i>Interface</i>	TTL UART, RXD, TXD
4	Komunikasi Saluran Listrik	220V 50/60Hz or low Voltage DC
No	Spesifikasi <i>Power Line Carrier</i>	Keterangan
5	Jarak Komunikasi	1500 meter (Tergantung pada keadaan lingkungan saluran listrik tersebut)
6	Panjang Frame	≤ 20 Bytes
7	Frekuensi Pembawa	290 KHz/125KHz
8	Kecepatan Data	500 bps
9	Modulasi	<i>Zeropassage bimodule / narrow band</i>
10	Suhu Kerja	-20 sampai 70 Derajat Celcius
11	Ukuran	4,8 cm X 3,1 cm X 1,3 cm (P x L x T)

2.3 Arduino IDE [4]



Gambar 2.3 Jendela Arduino IDE

Board Arduino dapat di program menggunakan *software open source* bawaan Arduino IDE. Arduino IDE adalah sebuah aplikasi *crossplatform* yang berbasis bahasa pemrograman *Processing* dan *Wiring*. Arduino IDE di desain untuk mempermudah pemrograman dengan adanya kode editor yang dilengkapi dengan *syntax highlighting*, *brace matching*, dan indentasi otomatis untuk kemudahan pembacaan program, serta dapat meng-*compile* dan meng-*upload* program ke *board* dalam satu klik. Jendela Arduino IDE dapat dilihat pada Gambar 2.3. IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. Editor program merupakan sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *Processing*.
2. *Compiler* merupakan sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah *microcontroller* tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh mikrokontroler adalah kode biner. Itulah sebabnya *compiler* diperlukan dalam hal ini.
3. *Uploader* merupakan sebuah modul yang memuat kode biner dari komputer ke dalam *memory* di dalam papan Arduino.

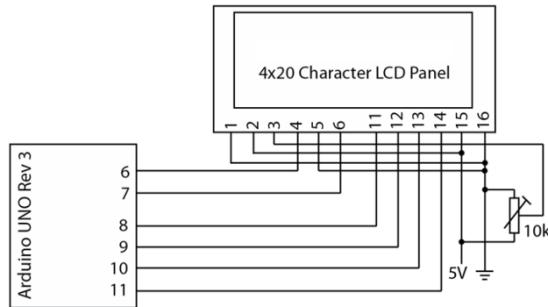
2.4 LCD (*Liquid Crystal Display*) [4]

LCD merupakan alat penampil karakter yang dapat berupa angka, huruf, dan gambar. Jenis-jenis LCD yang terdapat dipasaran ada dua jenis yaitu LCD teks dan LCD grafik. LCD yang digunakan adalah jenis LCD M1632. Modul M1632 merupakan modul LCD dengan tampilan 20x4 baris dengan konsumsi daya yang rendah. *LCD* membutuhkan *driver* supaya bisa dikoneksikan dengan sistem minimum dalam suatu mikrokontroler. *Driver* yang disebutkan berisi rangkaian pengaman, pengatur tingkat kecerahan maupun data, serta untuk mempermudah pemasangan di mikrokontroler. LCD Teks adalah jenis LCD yang digunakan untuk menampilkan teks atau angka dalam kode ASCII. Sedangkan *graphic* LCD dapat menampilkan gambar karena setiap sel dari grafik LCD memuat "dot" kristal cair. LCD banyak digunakan sebagai *display* dari alat – alat elektronika seperti kalkulator, multimeter digital, jam digital dan sebagainya. Skema rangkaian pin LCD 20 x 4 terdapat pada Gambar 2.4 Skema Rangkaian LCD 20 x 4. Konfigurasi pin yang dimiliki oleh LCD 20 x 4 terdapat pada Tabel 2.2 Konfigurasi Pin LCD 20x4.

Tabel 2.2 Konfigurasi Pin LCD 20x4

Fungsi	Pin	Deskripsi	Data Logika	Keterangan
Ground	1	Vss (Ground)		0 V
Tegangan Sumber	2	Vdd (Vcc)		+5 V
Kontras	3	VEE		
Kendali LCD	4	RS	0	
			1	
	5	R/W	0	
			1	
	6	E	0	
			1	
			Floating	
Data	7	D0	0/1	Bit
	8	D1	0/1	Bit
	9	D2	0/1	Bit
	10	D3	0/1	Bit
	11	D4	0/1	Bit
	12	D5	0/1	Bit

	13	D6	0/1	Bit
	14	D7	0/1	Bit
Anoda	15	<i>Backlight Lampu Background</i>		
Katoda	16	<i>Backlight Lampu Background</i>		

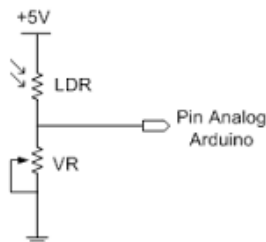


Gambar 2.4 Skema Rangkaian LCD 20 x 4

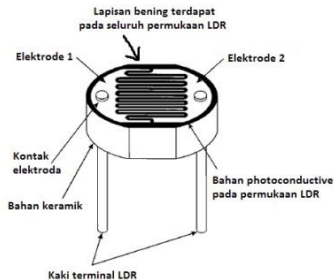
2.5 *Light Dependent Resistor* [5]

Light Dependent Resistor atau yang biasa disebut LDR adalah jenis resistor yang nilainya berubah seiring intensitas cahaya yang diterima oleh komponen tersebut. Biasa digunakan sebagai detektor cahaya atau pengukur besaran konversi cahaya. *Light Dependent Resistor*, terdiri dari sebuah cakram semikonduktor yang mempunyai dua buah elektroda pada permukaannya. Pada saat gelap atau cahaya redup, bahan dari cakram tersebut menghasilkan elektron bebas dengan jumlah yang relatif kecil. Sehingga hanya ada sedikit elektron untuk mengangkut muatan elektrik. Artinya pada saat cahaya redup LDR menjadi konduktor yang buruk, atau bisa disebut juga LDR memiliki resistansi yang besar pada saat gelap atau cahaya redup. Pada saat cahaya terang, ada lebih banyak elektron yang lepas dari atom bahan semikonduktor tersebut. Sehingga akan ada lebih banyak elektron untuk mengangkut muatan elektrik. Artinya pada saat cahaya terang LDR menjadi konduktor yang baik, atau bisa disebut juga LDR memiliki resistansi yang kecil pada saat cahaya terang. Bila sebuah LDR dibawa dari suatu ruangan dengan level kekuatan cahaya tertentu kedalam suatu ruangan yang gelap sekali, maka bisa kita amati bahwa nilai resistansi dari LDR tidak akan segera berubah

resistansinya pada keadaan ruangan gelap tersebut. Namun LDR tersebut hanya akan bisa mencapai harga dikegelapan setelah mengalami selang waktu tertentu. dan suatu kenaikan nilai resistansi dalam waktu tertentu. Harga ini ditulis dalam $K \Omega / \text{detik}$. Untuk LDR tipe arus harganya lebih besar dari $200 K \Omega / \text{detik}$ (selama 20 menit pertama mulai dari level cahaya 100 lux), kecepatan tersebut akan lebih tinggi pada arah sebaliknya, yaitu pindah dari tempat gelap ke tempat terang yang memerlukan waktu kurang dari 10 ms untuk mencapai resistansi yang sesuai dengan level cahaya 400 lux. Rangkaian LDR dapat dilihat pada Gambar 2.5 Skema Rangkaian *Light Dependent Resistor*. Dan konstruksi dari *light dependent resistor* dapat dilihat pada Gambar 2.6 Konstruksi *Light Dependent Resistor*.



Gambar 2.5 Skema Rangkaian *Light Dependent Resistor*



Gambar 2.6 Konstruksi *Light Dependent Resistor*

2.6 Push Button [5]

Saklar *push button* adalah tipe saklar yang hanya kontak sesaat saja saat ditekan dan setelah dilepas maka akan kembali lagi menjadi NO (*Normally Open*), biasanya saklar tipe NO ini memiliki rangkaian penguncinya yang dihubungkan dengan kontaktor dan tipe NO digunakan

untuk tombol *on*. *Push button* ada juga yang bertipe NC (*Normally Close*), biasanya digunakan untuk tombol *off*. Terdapat 4 konfigurasi saklar *push button*: a. Tanpa-pengunci (*no guard*), b. Pengunci-penuh (*full guard*), c. *Extended guard*, dan d. *Mushroom button*.

Cara kerja saklar *push button* alat ini befungsi sebagai pemberi sinyal masukan pada rangkaian listrik, ketika / selama bagian *knopnya* ditekan maka alat ini akan bekerja sehingga kontak-kontaknya akan terhubung untuk jenis *normally open* dan akan terlepas untuk jenis *normally close*, dan sebaliknya ketika *knopnya* dilepas kembali maka kebalikan dari sebelumnya, untuk membuktikannya pada terminalnya bisa digunakan alat ukur tester / Ohm meter. Pada umumnya pemakaian terminal jenis NO digunakan untuk menghidupkan rangkaian dan terminal jenis NC digunakan untuk mematikan rangkaian, namun semuanya tergantung dari kebutuhan. Rangkaian *push button* dapat dilihat pada Gambar 2.7 Skema Rangkaian *Push Button*.



Gambar 2.7 Skema Rangkaian *Push Button*

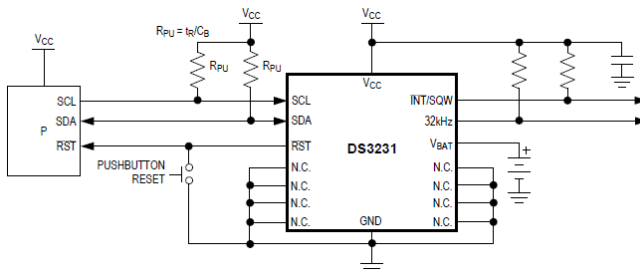
Dan M. Fatkhur Rozi membutuhkan teori penunjang dari *Real Time Clock (RTC)*, Memori., Arduino Mega, dan *Serial Peripheral Interface (SPI)*.

2.7 *Real Time Clock (RTC)* [4]

Real Time Clock (RTC) merupakan suatu *chip IC* yang memiliki fungsi untuk menyimpan informasi detik, menit, jam, hari, tanggal, bulan, dan tahun. Tanggal pada akhir bulan secara otomatis disesuaikan dengan bulan dengan kurang dari 31 hari, termasuk koreksi untuk tahun kabisat. Jam beroperasi baik dalam 24 jam atau format 12 jam dengan indikator AM/PM aktif ketika kondisi *low*. Sebuah *temperature* yang presisi dikompensasi sebagai tegangan referensi. Rangkaian komparator memonitor status VCC untuk mendeteksi gangguan listrik, untuk memberikan *output* ulang dan secara otomatis terhubung ke sumber cadangan jika diperlukan. Selain itu, pin RST akan aktif ketika tegangan 0 Volt yang dideteksi sebagai tombol tekan untuk menghasilkan perintah

reset.

Data-data yang tersimpan pada IC DS3231 disimpan pada *register* 00H untuk detik, 01H untuk menit, 02H untuk jam, 03H untuk hari, 04H untuk tanggal, 05H untuk bulan, 06H untuk tahun, 07H untuk kontrol dan RAM 56x8 pada *register* 08H-3FH. *Register* tersebut bisa diakses oleh mikrokontroler melalui komunikasi I2C. Skematik dari RTC



dapat dilihat pada Gambar 2.8.

Gambar 2.8 Skematik RTC DS3231

2.8 Memori [4]

Memori dapat dibagi menjadi dua yakni memori eksternal dan memori internal. Memori eksternal terdiri dari beberapa macam jenis memori, salah satunya yakni *Micro SD*. *Micro SD* adalah kartu memori *non-volatile* yang dikembangkan oleh *SD Card Association* yang digunakan dalam perangkat *portable*. Pada Gambar 2.9 *Memory Card* dapat dilihat konstruksi dari memori eksternal. Saat ini, teknologi *micro SD* sudah digunakan oleh lebih dari 400 *brand* produk serta dianggap sebagai standar *industri de-facto*. Keluarga *micro SD* yang lain terbagi menjadi SDSC yang kapasitas maksimum resminya sekitar 2 GB, meskipun beberapa ada yang sampai 4 GB. SDHC (*High Capacity*) memiliki kapasitas dari 4 GB sampai 32 GB. Dan SDXC (*Extended Capacity*) kapasitasnya di atas 32 GB hingga maksimum 2 TB.

Dari sudut pandang perangkat, semua kartu ini termasuk ke dalam keluarga SD. *SD adapter* memungkinkan konversi fisik kartu SD yang lebih kecil untuk bekerja di slot fisik yang lebih besar dan pada dasarnya ini adalah alat pasif yang menghubungkan pin dari mikro SD yang kecil ke pin adaptor mikro SD yang lebih besar. SD mempunyai bentuk fisik yang sama maka sering menyebabkan kebingungan di kalangan konsumen. Contohnya, *Micro SD*, *Micro SDHC*, dan *Micro SDXC* ukuran

fisiknya sama tetapi kapabilitasnya berbeda. Protokol komunikasi untuk SDHC/SDXC/SDIO sedikit berbeda dengan *Micro SD* yang sudah mapan karena biasanya *host device* keluaran lama tidak bisa mengenali kartu keluaran baru. Kebanyakan masalah mengenai *inkompatibilitas* ini dapat diselesaikan dengan *firmware update*.



Gambar 2.9 *Memory Card*

2.9 *Board Arduino Mega* [4]



Gambar 2.10 *Board Arduino Mega*

Arduino Mega2560 adalah papan mikrokontroler berbasis ATmega2560. Arduino Mega2560 memiliki 54 *pin digital input/output*, dimana 15 pin dapat digunakan sebagai *output PWM*, 16 pin sebagai *input analog*, dan 4 pin sebagai UART (*port serial hardware*), 16 MHz *kristal osilator*, koneksi USB, *jack power*, *header ICSP*, dan tombol *reset*. Bentuk fisik dari Arduino Mega2560 dapat dilihat pada Gambar 2.10 *Board Arduino Mega*. Ini semua yang diperlukan untuk mendukung mikrokontroler. Cukup dengan menghubungkannya ke komputer melalui kabel USB atau power dihubungkan dengan adaptor AC-DC atau baterai untuk mulai mengaktifkannya. Arduino Mega2560 kompatibel dengan sebagian besar *shield* yang dirancang untuk *Arduino Duemilanove* atau *Arduino Diecimila*.

Arduino Mega2560 memiliki fitur-fitur baru berikut :

1. *1.0 pin out* : Ditambahkan pin SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, IOREF memungkinkan *shield* untuk beradaptasi dengan tegangan yang tersedia pada papan.
2. Sirkuit RESET.
3. *Chip* ATmega16U2 menggantikan *chip* ATmega8U2.

Spesifikasi Arduino Mega2560 adalah sebagai berikut :

1. Mikrokontroler : ATMega 2560.
2. Tegangan Operasi : 5 V.
3. Tegangan *Input* : 7-12 V (yang direkomendasikan).
4. Tegangan *Input (min-maks)* : 6-20 V
5. Jumlah *Pin I/O Digital* : 54 (15 diantaranya merupakan *output PWM*).
6. Jumlah *Pin Analog Input* : 16.
7. Arus DC per *Pin I/O* : 40 mA.
8. Arus DC pada Pin 3,3V : 50 mA.
9. *Flash Memory* : 256 KB dimana 8 KB sebagai bootloader
10. *SRAM* : 8 KB.
11. *EPROM* : 4 KB.
12. *Clock Speed* : 16 MHz.

Arduino Mega dapat diaktifkan melalui koneksi USB atau dengan catu daya eksternal. Sumber daya dipilih secara otomatis. Sumber daya eksternal (non-USB) dapat berasal baik dari adaptor AC-DC atau baterai. Adaptor dapat dihubungkan dengan mencolokkan steker 2,1 mm yang bagian tengahnya terminal positif ke *jack* sumber tegangan pada papan. Jika tegangan berasal dari baterai dapat langsung dihubungkan melalui *header* pin Gnd dan pin Vin dari konektor POWER.

Papan Arduino ATmega2560 dapat beroperasi dengan pasokan daya eksternal 6 V sampai 20 V. Jika diberi tegangan kurang dari 7 V, maka, pin 5 V mungkin akan menghasilkan tegangan kurang dari 5 V dan ini akan membuat papan menjadi tidak stabil. Jika sumber tegangan menggunakan lebih dari 12 V, regulator tegangan akan mengalami panas berlebihan dan bisa merusak papan. Rentang sumber tegangan yang dianjurkan adalah 7 V sampai 12 V.

Pin tegangan yang tersedia pada papan Arduino adalah sebagai berikut :

1. VIN : Adalah *input* tegangan untuk papan Arduino ketika

menggunakan sumber daya *eksternal* (sebagai ‘saingan’ tegangan 5 V dari koneksi USB atau sumber daya *ter-regulator* lainnya). Dapat diberikan tegangan melalui pin ini, atau jika memasok tegangan untuk papan melalui *jack power*, bisa mengakses/mengambil tegangan melalui pin ini.

2. 5V : Sebuah pin yang mengeluarkan tegangan *ter-regulator* 5 V, dari pin ini tegangan sudah diatur (*ter-regulator*) dari regulator yang tersedia (*built-in*) pada papan. Arduino dapat diaktifkan dengan sumber daya baik berasal dari *jack power* DC (7-12 V), konektor USB (5 V), atau pin VIN pada *board* (7-12 V). Memberikan tegangan melalui pin 5 V atau 3,3 V secara langsung tanpa melewati regulator dapat merusak papan Arduino.
3. 3V3 : Sebuah pin yang menghasilkan tegangan 3,3 V. Tegangan ini dihasilkan oleh regulator yang terdapat pada papan (*on-board*). Arus maksimum yang dihasilkan adalah 50 mA.
4. GND : Pin *Ground* atau *Massa*.
5. IOREF : Pin ini pada papan Arduino berfungsi untuk memberikan referensi tegangan yang beroperasi pada mikrokontroler. Sebuah perisai (*shield*) dikonfigurasi dengan benar untuk dapat membaca pin tegangan IOREF dan memilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan (*voltage translator*) pada *output* untuk bekerja pada tegangan 5 V atau 3,3 V.

Arduino ATmega2560 memiliki 256 KB *flash memory* untuk menyimpan kode (yang 8 KB digunakan untuk *bootloader*), 8 KB SRAM dan 4 KB EEPROM (yang dapat dibaca dan ditulis dengan perpustakaan EEPROM).

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai *input* atau *output*, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Arduino Mega beroperasi pada tegangan 5 V. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up internal* (yang terputus secara *default*) sebesar 20-50 k Ω . Selain itu, beberapa pin memiliki fungsi khusus, antara lain :

1. Serial : 0 (RX) dan 1 (TX); Serial 1 : 19 (RX) dan 18 (TX);
Serial 2 : 17 (RX) dan 16 (TX); Serial 3 : 15 (RX) dan 14 (TX).

Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pins 0 dan 1 juga terhubung ke pin *chip* ATmega16U2 Serial USB-to-TTL.

2. *Eksternal Interupsi* : Pin 2 (*interrupt* 0), pin 3 (*interrupt* 1), pin 18 (*interrupt* 5), pin 19 (*interrupt* 4), pin 20 (*interrupt* 3), dan pin 21 (*interrupt* 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau perubah nilai.
3. SPI : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan *header* ICSP, yang secara fisik kompatibel dengan *Arduino Uno*, *Arduino Duemilanove* dan *Arduino Diecimila*.
4. LED : Pin 13. Tersedia secara *built-in* pada papan *Arduino ATmega2560*. LED terhubung ke pin digital 13. Ketika pin diset bernilai *HIGH*, maka LED menyala (*ON*), dan ketika pin diset bernilai *LOW*, maka LED padam (*OFF*).
5. TWI : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan *Wire*. Perhatikan bahwa pin ini tidak di lokasi yang sama dengan pin TWI pada *Arduino Duemilanove* atau *Arduino Diecimila*.

Arduino Mega2560 memiliki 16 pin sebagai *analog input*, yang masing-masing menyediakan resolusi 10 bit (yaitu 1024 nilai yang berbeda). Secara *default* pin ini dapat diukur/diatur dari mulai *Ground* sampai dengan 5 V, juga memungkinkan untuk mengubah titik jangkauan tertinggi atau terendah mereka menggunakan pin AREF dan fungsi *analogReference()*. Ada beberapa pin lainnya yang tersedia, antara lain :

1. AREF : Referensi tegangan untuk *input analog*. Digunakan dengan fungsi *analogReference()*.
2. RESET : Jalur *LOW* ini digunakan untuk *me-reset* (menghidupkan ulang) mikrokontroler. Jalur ini biasanya digunakan untuk menambahkan tombol *reset* pada *shield* yang menghalangi papan utama *Arduino*.

Arduino Mega2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan *Arduino* lain, atau dengan mikrokontroler lainnya. *Arduino ATmega328* menyediakan 4 *hardware* komunikasi serial UART TTL (5 V). Sebuah *chip* ATmega16U2

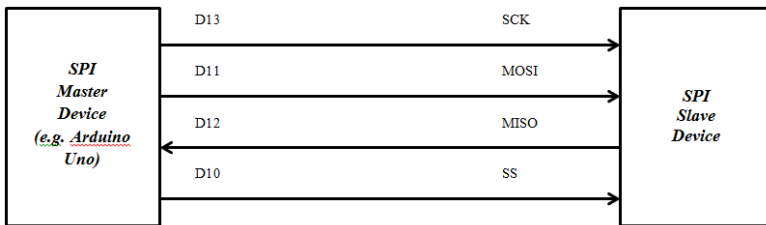
(ATmega8U2 pada papan Revisi 1 dan Revisi 2) yang terdapat pada papan digunakan sebagai media komunikasi serial melalui USB dan muncul sebagai *COM Port Virtual* (pada *Device* komputer) untuk berkomunikasi dengan perangkat lunak pada komputer, untuk sistem operasi Windows masih tetap memerlukan file ini, tetapi untuk sistem operasi OS X dan Linux akan mengenali papan sebagai port COM secara otomatis. Perangkat lunak Arduino termasuk di dalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui *chip* USB-to-serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1).

Sebuah perpustakaan *Software Serial* memungkinkan untuk komunikasi serial pada salah satu pin digital Mega2560. ATmega2560 juga mendukung komunikasi TWI dan SPI. Perangkat lunak Arduino termasuk perpustakaan *Wire* digunakan untuk menyederhanakan penggunaan bus TWI. Untuk komunikasi SPI, menggunakan perpustakaan SPI.

2.10 *Serial Peripheral Interface (SPI) [4]*

Serial Peripheral Interface (SPI) merupakan salah satu mode komunikasi serial *synchronous* kecepatan tinggi yang dimiliki oleh ATmega 328. Komunikasi SPI membutuhkan 3 jalur yaitu MOSI, MISO, dan SCK. Melalui komunikasi ini data dapat saling dikirimkan baik antara *mikrokontroller* maupun antara *mikrokontroller* dengan *peripheral* lain di luar *mikrokontroller*. Blok diagram *Serial Peripheral Interface* (SPI) dapat dilihat pada Gambar 2.11 Blok Diagram SPI.

1. MOSI (*Master Output Slave Input*)
Artinya jika dikonfigurasi sebagai *Master* maka pin MOSI sebagai *output* tetapi jika dikonfigurasi sebagai *Slave* maka pin MOSI sebagai *input*.
2. MISO (*Master Input Slave Output*)
Artinya jika dikonfigurasi sebagai *Master* maka pin MISO sebagai *input* tetapi jika dikonfigurasi sebagai *Slave* maka pin MISO sebagai *output*.
3. CLK (*Clock*)
Jika dikonfigurasi sebagai *Master* maka pin CLK berlaku sebagai *output* tetapi jika dikonfigurasi sebagai *Slave* maka pin CLK berlaku sebagai *input*.



Gambar 2.11 Blok Diagram SPI

Juga Biner Rochmatul C.A membutuhkan teori penunjang dari Arduino Uno.

2.11 Board Arduino Uno [6]

Arduino Uno adalah papan mikrokontroler berdasarkan ATmega328 (*datasheet*). Ini memiliki 14 *digital pin input / output* (dimana 6 dapat digunakan sebagai *output PWM*), 6 *input analog*, resonator keramik 16 MHz, koneksi USB, *jack* listrik, *header ICSP*, dan tombol *reset*. Ini berisi semua yang diperlukan untuk mendukung mikrokontroler; hanya menghubungkannya ke komputer dengan kabel USB atau *power* itu dengan adaptor AC-DC atau baterai untuk memulai menggunakannya. Bentuk fisik dari Arduino Uno dapat dilihat pada Gambar 2.12 *Board Arduino Uno*.

Uno berbeda dari semua papan sebelumnya di bahwa itu tidak menggunakan *chip driver FTDI USB-to-serial*. Sebaliknya, fitur Atmega16U2 (Atmega8U2 sampai versi R2) diprogram sebagai konverter *USB-to-serial*. Revisi ke 2 Uno memiliki resistor menarik garis 8U2 HWB *line to ground*, sehingga lebih mudah untuk dimasukkan ke dalam mode DFU. Revisi ke 3 memiliki fitur-fitur baru berikut: 1,0 *pinout*: menambahkan SDA dan pin SCL yang dekat dengan pin AREF dan dua pin baru lainnya ditempatkan dekat dengan pin RESET, yang IOREF yang memungkinkan perisai untuk beradaptasi dengan tegangan yang tersedia dari papan. Perisai akan kompatibel dengan kedua papan yang menggunakan AVR yang beroperasi dengan 5V dan dengan *Arduino Due* yang beroperasi dengan 3,3V. Yang kedua adalah pin tidak terhubung, yang disediakan untuk perkembangan mendatang. *Stronger RESET sirkuit*. Atmega 16U2 menggantikan 8U2.

"*Uno*" berarti satu di Italia dan diberi nama untuk menandai peluncuran Arduino 1.0. *The Uno* dan versi 1.0 akan menjadi versi referensi Arduino, bergerak maju. *The Uno* adalah yang terbaru dalam serangkaian USB Arduino papan, dan model referensi untuk *platform* Arduino.



Gambar 2.12 *Board* Arduino Uno

-----*Halaman ini sengaja dikosongkan*-----

BAB III

PERANCANGAN ALAT

Pada bab ini akan dibahas mengenai perancangan dan pembuatan alat “Pengukuran Metering KWh Pada Pelanggan Tegangan Rendah Dengan Media *Power Line Carrier*” yang dilakukan oleh Biner Rochmatul C.A dan M. Fatkhur Rozi. Perancangan alat ini dibagi menjadi tiga tahapan yakni, perancangan sistem keseluruhan, perancangan perangkat keras (*hardware*) yang meliputi perancangan *prototype* simulasi, perancangan KWh meter, perancangan *Master* serta perancangan *Slave*. Dan perancangan perangkat lunak (*software*) meliputi pemrograman arduino IDE pada arduino Uno untuk *Slave* serta arduino mega untuk *Master*.

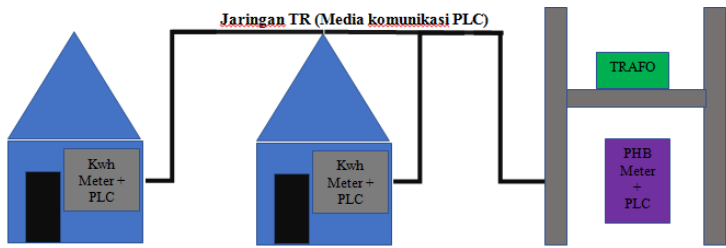
3.1 Perancangan Sistem Keseluruhan

Perancangan sistem secara keseluruhan dalam pembuatan alat yang digunakan oleh Biner Rochmatul C.A dan M. Fatkhur Rozi diperlihatkan pada Gambar 3.1.

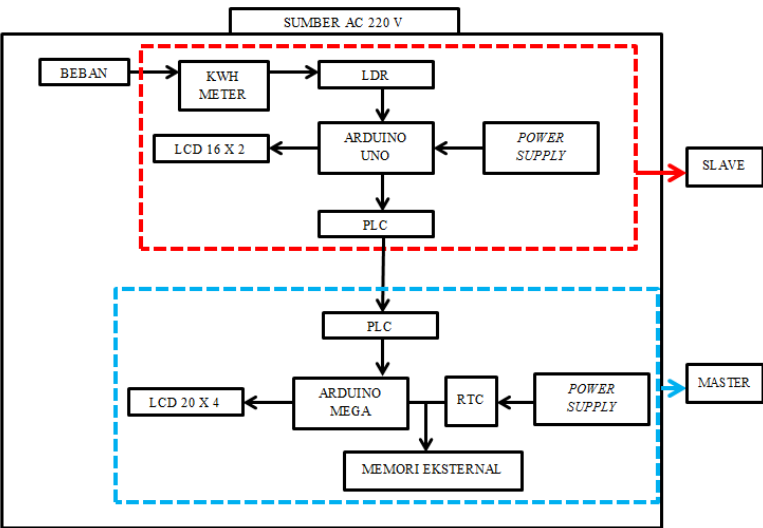
Perancangan sistem secara keseluruhan dalam pembuatan alat yang digunakan oleh Biner Rochmatul C.A dan M. Fatkhur Rozi diperlihatkan dengan pembagian *Slave* untuk Biner Rochmatul C.A dan *Master* untuk M. Fatkhur Rozi pada Gambar 3.2 Diagram Fungsional Alat Keseluruhan.

Dari Gambar 3.1 dan Gambar 3.2 dapat dipahami bahwa sistem perhitungan *impuls* yang digunakan untuk mendapatkan nilai data penggunaan kWh pelanggan. Setiap penggunaan beban yang terdapat pada pelanggan jaringan tegangan rendah dihitung oleh KWh meter digital. *Impuls* yang dihasilkan oleh KWh meter tersebut lalu ditangkap oleh sensor LDR (*Light Dependent Resistor*). Karena KWh meter yang digunakan memiliki tetapan 3200 *impuls*/kWh maka *impuls* yang tertangkap pada sensor LDR dihitung hingga sebanyak 3200 *impuls* yang setara dengan 1 kWh. Data *impuls* tersebut disimpan pada EEPROM yang terdapat pada arduino Uno. Data tersebut ditampilkan pada LCD 16 x 2. Ketika tepat pada waktu yang ditentukan, data kWh tersebut dikirim melalui media PLC yang terdapat pada *Slave* yang terpasang pada KWh meter pelanggan kepada *Master* melalui media PLC yang bertugas sebagai penghimpun data kWh pelanggan. Data kWh yang diterima oleh PLC yang terdapat pada *Master* akan disimpan oleh EEPROM yang terdapat pada arduino Mega. Data kWh tersebut juga akan ditampilkan

pada LCD 20 x 4. Data kWh yang dikirim oleh *Slave* ketika diterima oleh *Master* akan tercatat dan tersimpan pada kartu memori sesuai dengan waktu yang telah disesuaikan dengan RTC. RTC berfungsi sebagai penunjuk dan pencatat waktu penerimaan data kWh yang dikirim oleh *Slave* kepada *Master*.



Gambar 3.1 Rancangan Sistem Alat



Gambar 3.2 Diagram Fungsional Alat Keseluruhan

3.2 Perancangan *Hardware*

Pada perancangan *hardware* pada tugas akhir ini yang dibahas terdiri dari perancangan *prototype* simulator, perancangan KWh meter,

perancangan *Master* serta perancangan *Slave* yang merupakan perancangan elektronik.

Perancangan *hardware* dalam pembuatan alat yang digunakan oleh Biner Rochmatul C.A dan M. Fatkhur Rozi diantaranya yakni pada Sub Bab 3.2.1 Perancangan Prototype Simulator dan 3.2.2 Perancangan KWh Meter

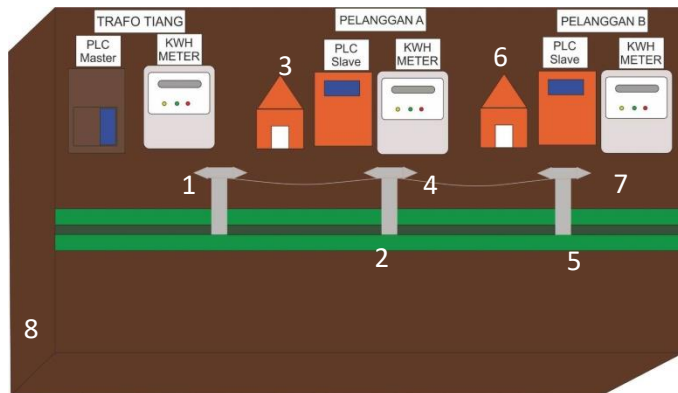
Perancangan *hardware* dalam pembuatan alat yang digunakan oleh Biner Rochmatul C.A yakni pada Sub Bab 3.2.3 Perancangan *Slave*.

Perancangan *hardware* dalam pembuatan alat yang digunakan oleh M. Fatkhur Rozi yakni pada Sub Bab 3.2.4 Perancangan *Master*.

Perancangan *hardware* dalam pembuatan alat yang digunakan oleh Biner Rochmatul C.A dan M. Fatkhur Rozi diantaranya yakni :

3.2.1 Perancangan *Prototype Simulator*

Rancangan *Prototype* simulator berikut merupakan rancangan akhir dari sistem yang bekerja secara keseluruhan.



Gambar 3.3 Perancangan *Prototype Simulator*

Tata letak *Prototype Simulator* dan peletakan alat Tugas Akhir dapat dilihat pada Gambar 3.3. Keterangan tata letak *Prototype Simulator* pada Gambar 3.3 adalah sebagai berikut :

1. Merupakan bagian dari KWh Meter yang terpasang pada *Prototype* panel PHB milik PT PLN.
2. Merupakan *prototype* Jaringan Tegangan Rendah untuk

pelanggan A. Juga terdapat beban resistif berupa lampu sebanyak dua buah dan masing-masing bernilai 100 W untuk Pelanggan A.

3. Merupakan *prototype* Rumah Pelanggan Jaringan Tegangan Rendah A.
4. Merupakan *prototype* KWh Meter dan *Slave* PIC Pelanggan A.
5. Merupakan *prototype* Jaringan Tegangan Rendah untuk pelanggan B. Juga terdapat beban resistif berupa lampu sebanyak dua buah dan masing-masing bernilai 100 W untuk Pelanggan B.
6. Merupakan *prototype* Rumah Pelanggan Jaringan Tegangan Rendah A.
7. Merupakan *prototype* KWh Meter dan *Slave* PIC Pelanggan B.
8. Merupakan *Wiring Room*. Ruangan yang berisi dari berbagai macam *wiring* KWh Meter dan beban.

Perancangan *Prototype* Simulator berupa *prototype* dari rumah-rumah pelanggan PT PLN jaringan tegangan rendah dan panel PHB milik PT PLN. Pemodelan Perancangan *Prototype* Simulator ini berbahan dasar triplek yang berbentuk lemari papan persegi panjang dengan ukuran 125 cm X 75 cm yang dibagi menjadi tiga bagian yakni panel PHB milik PT PLN, Pelanggan Jaringan Tegangan Rendah A, dan Pelanggan Jaringan Tegangan B. Pada bagian tampak depan terdapat 3 KWh Meter sebagai bentuk perlambangan dari KWh Meter panel PHB milik PLN, KWh Meter Pelanggan Jaringan Tegangan Rendah A, dan KWh Meter Pelanggan Jaringan Tegangan B. Pada bagian depan rumah Pelanggan Jaringan Tegangan Rendah A dan Pelanggan Jaringan Tegangan B terdapat beberapa tiang listrik yang merupakan beban resistif berupa lampu bohlam 100 W yang diasumsikan sebagai beban yang biasa digunakan oleh para pelanggan jaringan tegangan rendah. Pada tepat bagian bawah KWh Meter Pelanggan Jaringan Tegangan Rendah A dan Pelanggan Jaringan Tegangan B terdapat *Slave* yang berfungsi untuk mengirim dan mengolah *impuls* yang dihasilkan oleh KWh Meter pelanggan menjadi nilai kWh dan mengirimkannya pada *Master* yang terdapat pada panel PHB milik PT PLN. *Slave* merupakan serangkaian kontroler pada KWh Meter pelanggan dari Tugas Akhir ini. Semua rangkaian yang dibuat terdiri dari rangkaian LCD 16 x 2, sensor *Light Dependent Resistor* (LDR), Arduino Uno, rangkaian *Power Supply*,

Power Line Carrier, dan LM3805 , dikemas menjadi 1 pada bagian tepat dibawah KWh Meter Pelanggan.

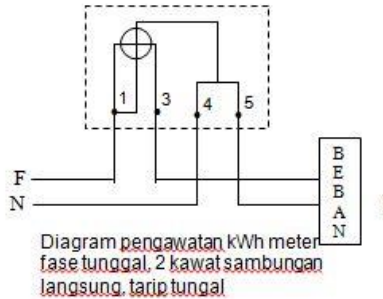
Tepat dibawah KWh Meter pada panel PHB PT PLN juga terdapat *Master*. *Master* berfungsi untuk menerima dan mengolah kWh guna disimpan pada EEPROM Arduino Mega. Kemudian ditampilkan pada LCD 20 x 4. Data penerimaan serta pengolahan kWh oleh Arduino Mega tersebut dicatat dan disimpan pada memori sesuai dengan waktu yang ditunjukkan pada RTC yakni setiap pada pukul 10.00. *Master* merupakan serangkaian kontroler pada KWh Meter pada panel PHB dari Tugas Akhir ini. Semua rangkaian yang dibuat terdiri dari rangkaian LCD 20 x 4, sensor *Light Dependent Resistor* (LDR), Arduino Mega, *Real Time Clock* (RTC), Memori, rangkaian *Power Supply*, *Power Line Carrier*, dan LM3805 , dikemas menjadi 1 pada bagian tepat dibawah KWh Meter pada panel PHB.

Pada bagian belakang yakni tepatnya didalam lemari disebut dengan *wiring room*. *Wiring Room* merupakan tempat dimana berbagai macam *wiring* KWh Meter pelanggan dan KWh Meter yang terletak pada panel PHB serta *Wiring* beban.

3.2.2 Perancangan KWh Meter

Perancangan KWh Meter ini digunakan untuk mengukur kWh yang telah digunakan oleh pelanggan. KWh yang digunakan yakni KWh Meter elektronik / digital. Sehingga pengukuran nilai kWh dengan KWh Meter ini dapat menggunakan perhitungan *impuls*. Pada KWh Meter yang kami gunakan ini memiliki tetapan yakni 3200 *impuls*/KWh. Dimana berarti bawa ketika *impuls* pada KWh Meter ini telah berkedip sebanyak 3200 kali, maka akan bernilai sama dengan 1 KWh. Perancangan *Wiring* KWh Meter ini dapat dilihat pada Gambar 3.4 *Wiring* KWh Meter.

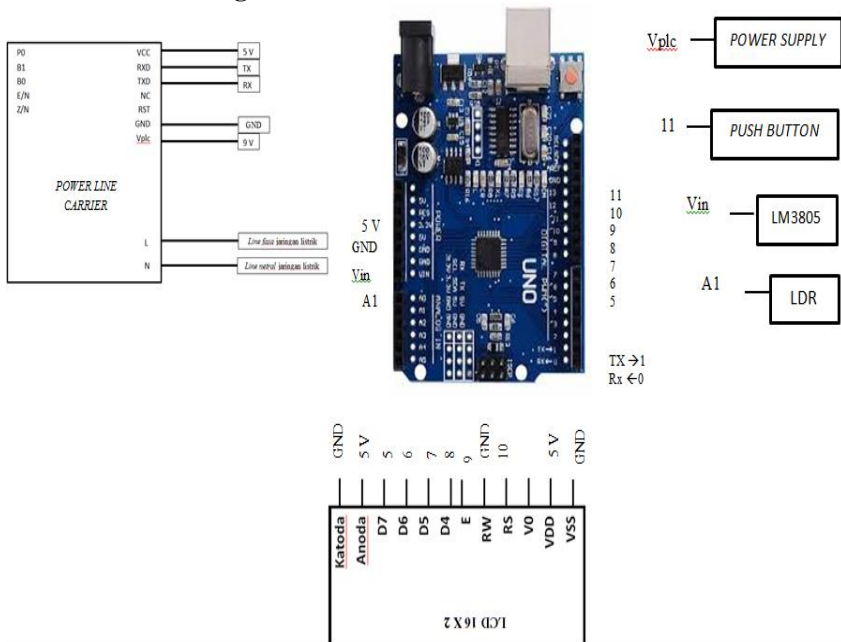
Perancangan KWh Meter ini meliputi *wiring* pada KWh. Fasa 220V terhubung pada saluran fasa KWh dan terhubung dengan fasa beban. Sedangkan saluran netral terhubung dengan netral KWh serta netral beban.



Gambar 3.4 Wiring KWh Meter

Perancangan *hardware* dalam pembuatan alat yang digunakan oleh Biner Rochmatul C.A diantaranya yakni :

3.2.3 Perancangan *Slave*



Gambar 3.5 Perancangan *Slave*

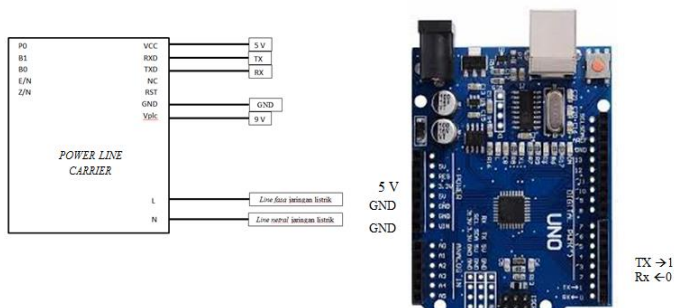
Perancangan *Slave* ini berfungsi untuk mendukung KWh meter dalam mengolah serta mengukur kWh yang telah digunakan oleh pelanggan. Perancangan *Slave* yakni perancangan komponen pengirim data KWh yang ada pada pelanggan. *Slave* terdiri dari beberapa komponen diantaranya yakni PLC, LCD, *Power Supply*, LM3805, LDR, serta arduino Uno. Perancangan *Slave* ini dapat dilihat pada Gambar 3.5 Perancangan *Slave*.

Karena terdiri dari berbagai macam komponen tersebut, maka *Slave* juga terbagi menjadi beberapa perancangan komponen penyusunnya. Beberapa komponen ini nantinya akan dihubungkan dengan arduino Uno.

3.2.3.1 Perancangan *Power Line Carrier* (PLC)

Pengukuran nilai kWh pelanggan yang dilakukan oleh KWh Meter pelanggan akan memperoleh data penggunaan kWh oleh pelanggan khususnya jaringan tegangan rendah. Untuk penyaluran data dari KWh meter rumah pelanggan tersebut terhadap KWh meter pada KWh meter penghimpun data maka digunakan media *Power Line Carrier* (PLC). Media komunikasi PLC ini digunakan dalam jaringan PLN 220 V. Perancangan komunikasi ini dapat dilihat pada Gambar 3.6 Perancangan *Power Line Carrier*. Media Komunikasi ini akan digunakan baik disisi pelanggan sebagai pengirim data maupun dari sisi KWh Meter penghimpun data kWh pada sisi panel PHB. Berikut adalah sambungan antara PLC dengan Arduino Uno :

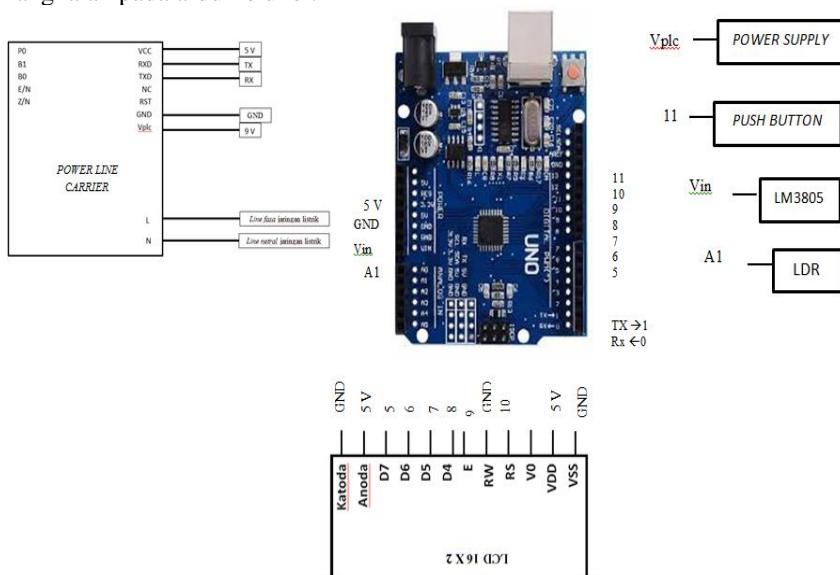
1. Tx – Pin Rx Arduino Uno
2. Rx – Pin Tx Arduino Uno
3. VCC – Pin 5 Volt Arduino Uno
4. GND – Pin *Ground* Arduino Uno



Gambar 3.6 Perancangan *Power Line Carrier* dengan Arduino Uno

1. V PLC – *Line Power Supply*

Arduino Uno merupakan *control system* dari *Slave* sendiri. Arduino Uno berfungsi sebagai pengolah data kWh pelanggan dari pengukuran *impuls* yang diperoleh dari KWh meter pelanggan. Data kWh pelanggan juga disimpan pada EEPROM arduino Uno sebelum dikirim pada *Master* melalui media *Power Line Carrier*. Data kWh yang telah diolah oleh arduino Uno juga akan ditampilkan pada LCD 16 x 2 yang terhubung pada arduino uno ini. Pada perancangan *Slave* ini, arduino uno terhubung dengan LDR, PLC, dan LCD. Model perancangan ini dapat dilihat pada Gambar 3.7 Perancangan Arduino Uno. Berikut merupakan rangkaian pada arduino uno :

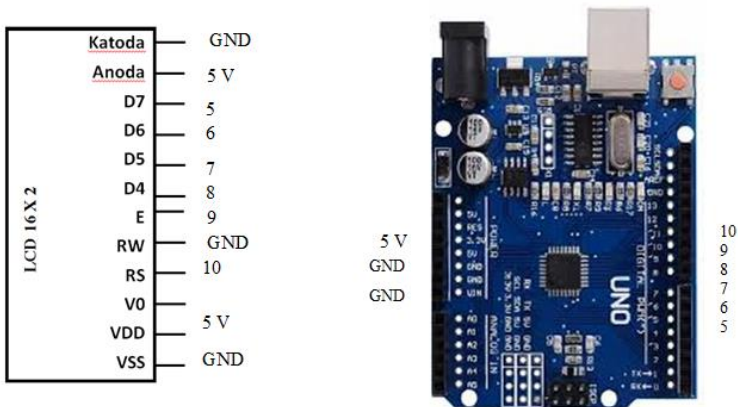


Gambar 3.7 Perancangan Arduino Uno

LCD yang digunakan pada *Slave* yakni LCD 16 x 2. Berfungsi untuk menampilkan data kWh yang terukur pada KWh meter pelanggan. Data kWh meter pelanggan yang terukur oleh KWh meter tersebut lalu

diolah oleh arduino uno dan disimpan pada EEPROM. Data kWh tersebut kemudian ditampilkan pada LCD sesuai dengan kWh yang terukur pada saat itu. Perancangan konfigurasi antara *pin* arduino uno dengan LCD dapat dilihat pada Gambar 3.8 Perancangan LCD dengan Arduino Uno. Berikut adalah sambungan antara LCD dengan Arduino Uno :

1. *Katoda* – Pin GND Arduino Uno
2. *Anoda* – Pin 5 V Arduino Uno
3. D7 – Pin 5 Arduino Uno
4. D6 – Pin 6 Arduino Uno
5. D5 – Pin 7 Arduino Uno
6. D4 – Pin 8 Arduino Uno
7. E – Pin 9 Arduino Uno
8. RW – Pin GND Arduino Uno
9. RS – Pin 10 Arduino Uno
10. V0 – Keluaran Potensio
11. VDD – Pin 5V Arduino Uno
12. VSS – Pin GND Arduino Uno



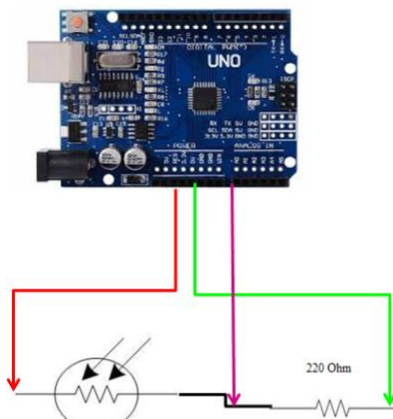
Gambar 3.8 Perancangan LCD dengan Arduino Uno

3.2.3.4 Perancangan *Light Dependent Resistor* (LDR)

Light Dependent Resistor merupakan sensor yang digunakan sebagai pendeteksi *impuls* yang dihasilkan oleh KWh Meter digital setelah terjadinya pemakaian daya pada beban yang digunakan. LDR akan mendeteksi kedip dari *impuls* KWh meter, yang mana banyaknya berbanding lurus dengan penggunaan daya oleh beban. Nilai pembacaan

dari nyala LDR ini dipengaruhi oleh nyala dari lampu indikator *impuls* pada KWh Meter. Perancangan konfigurasi antara *pin* arduino uno dengan LDR dapat dilihat pada Gambar 3.9 Perancangan *Light Dependent Resistor* dengan Arduino Uno. Berikut adalah sambungan antara *Light Dependent Resistor* dengan Arduino Uno :

1. *Output* – Pin A1 Arduino Uno
2. *VCC* – Pin 5 Volt Arduino Uno
3. *GND* – Pin *Ground* Arduino Uno



Gambar 3.9 Perancangan *Light Dependent Resistor* dengan Arduino Uno

3.2.3.5 Perancangan LM 3805

LM 3805 Berfungsi sebagai regulator tegangan yang dibutuhkan oleh arduino. Yang berfungsi sebagai penstabil tegangan. Berikut adalah sambungan antara LM3805 dengan Arduino Uno :

1. *Output* – Pin Vin Arduino Uno

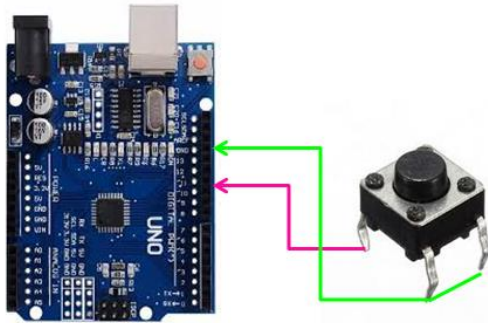
Berikut adalah sambungan antara LM3805 dengan *Power Supply* :

1. *Input* – *Output power supply*
2. *GND* – *GND Power Supply*

3.2.3.6 Perancangan *Push Button*

Push Button berfungsi sebagai tombol untuk melihat data yang terinput pada *LCD Slave*. Perancangan konfigurasi antara *pin* arduino uno dengan *Push Button* dapat dilihat pada Gambar 3.10 Perancangan *Push Button* dengan Arduino Uno. Berikut adalah sambungan antara *Push Button* dengan Arduino Uno :

1. Pin 1 dan Pin 2 – Pin 11 Arduino Uno
2. Pin 3 dan Pin 4 – GND Arduino Uno



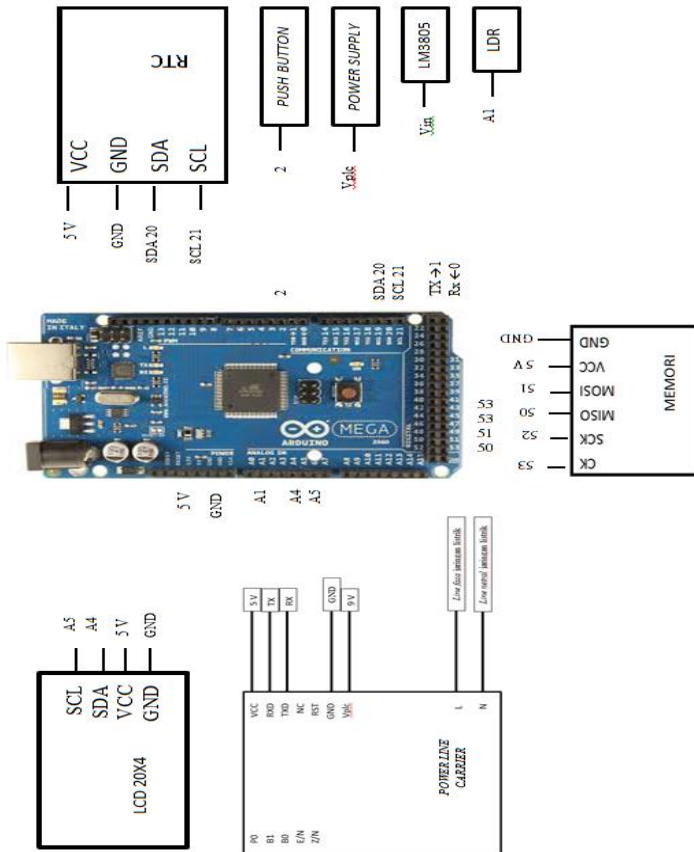
Gambar 3.10 Perancangan *Push Button* dengan Arduino Uno

Perancangan *hardware* dalam pembuatan alat yang digunakan oleh M. Fatkhur Rozi diantaranya yakni :

3.2.4 Perancangan *Master*

Perancangan *Master* ini berfungsi untuk mendukung KWh meter dalam mengolah serta mengukur kWh yang telah digunakan oleh para pelanggan. perancangan *Master* ini membantu dalam hal menghimpun data penggunaan kWh oleh pelanggan. Perancangan *Master* yakni perancangan komponen penerima data yang ada pada pelanggan. *Master* terdiri dari beberapa komponen diantaranya yakni PLC, Memori, RTC, LCD, *Power Supply*, LM3805, LDR, serta arduino Mega. Perancangan *Master* ini dapat dilihat pada Gambar 3.11 Perancangan *Master*.

Karena terdiri dari berbagai macam komponen tersebut, maka *Master* juga terbagi menjadi beberapa perancangan komponen penyusunnya. Beberapa komponen ini nantinya akan dihubungkan dengan arduino Mega.



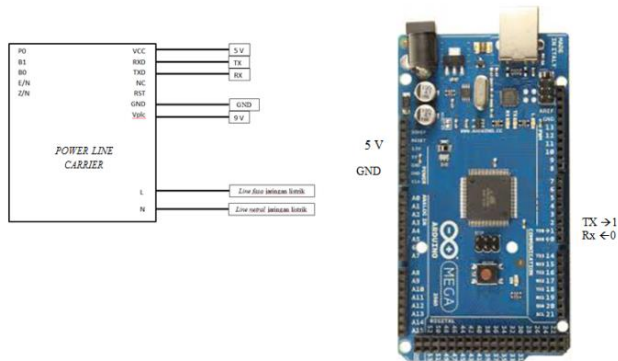
Gambar 3.11 Perancangan *Master*

3.2.4.1 Perancangan *Power Line Carrier* (PLC)

Pengukuran nilai kWh pelanggan yang dilakukan oleh KWh Meter pelanggan akan memperoleh data penggunaan kWh oleh pelanggan khususnya jaringan tegangan rendah. Untuk penyaluran data dari KWh meter rumah pelanggan tersebut terhadap KWh meter pada KWh meter penghimpun data maka digunakan media *Power Line Carrier* (PLC). Media komunikasi PLC ini digunakan dalam jaringan PLN 220 V. Media Komunikasi ini akan digunakan baik disisi pelanggan sebagai pengirim

data maupun dari sisi KWh Meter penghimpun data kWh pada sisi panel PHB. Untuk penyaluran data dari KWh meter rumah pelanggan terhadap KWh meter pada KWh meter penghimpun data maka digunakan media PLC. Perancangan konfigurasi antara *pin* arduino mega dengan PLC dapat dilihat pada Gambar 3.12 Perancangan *Power Line Carrier* dengan Arduino Mega. Berikut adalah sambungan antara *PLC* dengan Arduino Mega :

1. Tx – Pin Rx Arduino Mega
2. Rx – Pin Tx Arduino Mega
3. VCC – Pin 5 Volt Arduino Mega
4. GND – Pin *Ground* Arduino Mega



Gambar 3.12 Perancangan *Power Line Carrier* dengan Arduino Mega

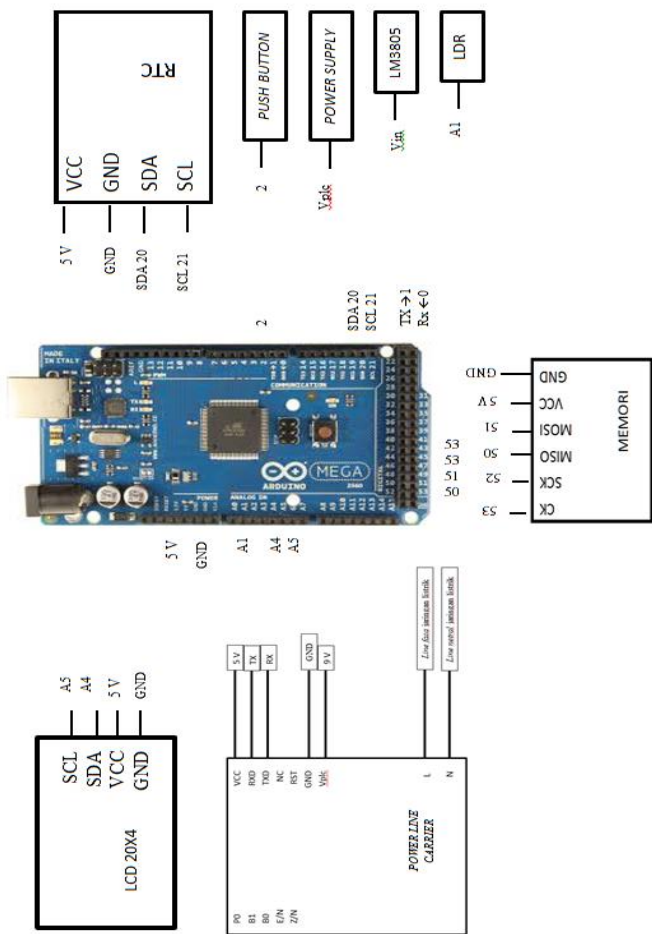
Berikut adalah sambungan antara *PLC* dengan *Power Supply*:

1. V PLC – *Line Power Supply*

3.2.4.2 Perancangan Arduino Mega

Arduino Mega merupakan *control system* dari *Master* sendiri. Arduino Mega berfungsi sebagai pengolah data kWh pelanggan-pelanggan jaringan tegangan rendah yang dikirim oleh *Slave*. Data kWh para pelanggan juga disimpan pada EEPROM arduino Mega sebelum dicatat pada memori yang terdapat pada *Master*. Data kWh yang telah diolah oleh arduino Mega juga akan ditampilkan pada LCD 20 x 4 yang terhubung pada arduino mega ini. Pada perancangan *Master* ini, arduino mega terhubung dengan LDR, PLC, Memori, RTC, dan LCD. Perancangan konfigurasi antara *pin* arduino mega dengan LDR, PLC, Memori, RTC,

dan LCD dapat dilihat pada Gambar 3.13 Perancangan Arduino Mega. Berikut merupakan rangkaian pada arduino mega :



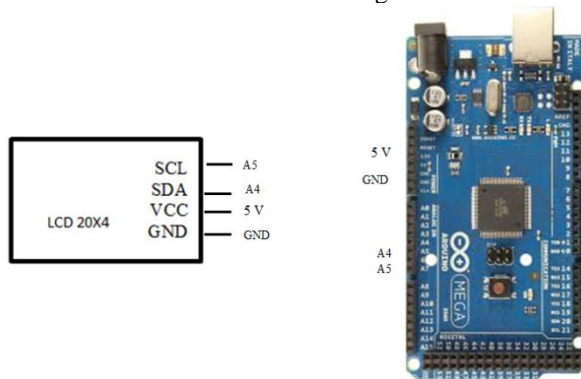
Gambar 3.13 Perancangan Arduino Mega

3.2.4.3 Perancangan LCD

LCD yang digunakan pada *Master* yakni LCD 20 x 4. Berfungsi untuk menampilkan data yang terukur pada KWh meter pelanggan yang terhimpun pada panel PHB. Data kWh meter pelanggan

yang terukur oleh KWh meter tersebut lalu dihimpun oleh *Master* pada panel PHB. Data kWh yang terkumpul tersebut lalu disimpan pada EEPROM. Data kWh tersebut kemudian ditampilkan pada LCD sesuai dengan kWh yang terukur pada saat itu. Perancangan konfigurasi antara *pin* arduino mega dengan LCD dapat dilihat pada Gambar 3.14 Perancangan LCD dengan Arduino Mega. Berikut adalah sambungan antara LCD dengan Arduino Mega :

1. SCL – Pin A5 Arduino Mega
2. SDA – Pin A4 Arduino Mega
3. VCC – Pin 5 Volt Arduino Mega
4. GND – Pin *Ground* Arduino Mega

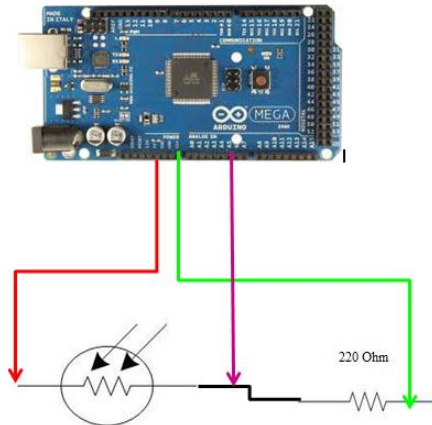


Gambar 3.14 Perancangan LCD dengan Arduino Mega

3.2.4.4 Perancangan *Light Dependent Resistor* (LDR)

Light Dependent Resistor merupakan sensor yang digunakan sebagai pendeteksi *impuls* yang dihasilkan oleh KWh Meter digital setelah terjadinya pengukuran daya pada beban yang digunakan. LDR akan mendeteksi kedip dari *impuls* kWh, sebanyak dari penggunaan daya oleh beban. Nilai dari nyala LDR ini dipengaruhi oleh nyala dari lampu indikator *impuls* pada KWh Meter. Perancangan konfigurasi antara *pin* arduino mega dengan LDR dapat dilihat pada Gambar 3.15 Perancangan *Light Dependent Resistor* dengan Arduino Mega. Berikut adalah sambungan antara *Light Dependent Resistor* dengan Arduino Mega :

1. *Output* – Pin A1 Arduino Mega
2. VCC – Pin 5 Volt Arduino Mega
3. GND – Pin *Ground* Arduino Mega



Gambar 3.15 Perancangan *Light Dependent Resistor* dengan Arduino Mega

3.2.4.5 Perancangan LM 3805

LM 3805 Berfungsi sebagai regulator tegangan yang dibutuhkan oleh arduino. Berikut adalah sambungan antara LM3805 dengan Arduino Mega :

1. *Output* – Pin Vin Arduino Mega

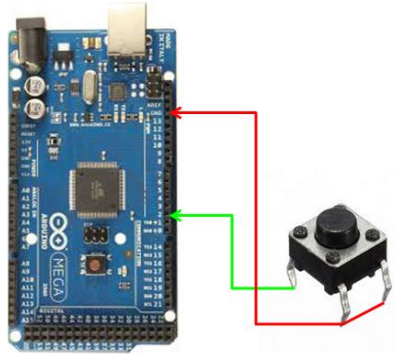
Berikut adalah sambungan antara LM3805 dengan *Power Supply* :

1. *Input* – *Output Power supply*
2. GND – GND *Power Supply*

3.2.4.6 Perancangan *Push Button*

Push Button berfungsi sebagai tombol untuk melihat data yang terinput pada LCD *Master*. Perancangan konfigurasi antara *pin* arduino mega dengan *Push Button* dapat dilihat pada Gambar 3.16 Perancangan *Push Button* dengan Arduino Mega. Berikut adalah sambungan antara *Push Button* dengan Arduino Mega :

1. Pin 1 dan Pin 2 – Pin 2 Arduino Mega
2. Pin 3 dan Pin 4 – GND Arduino Mega

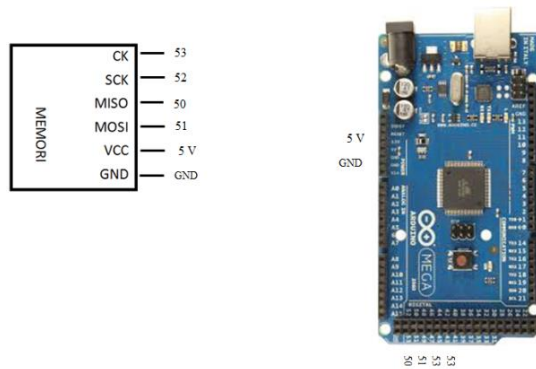


Gambar 3.16 Perancangan *Push Button* dengan Arduino Mega

3.2.4.7 Perancangan Memori Eksternal

Untuk pencatatan data diperlukan memori untuk menyimpan data yang telah terbaca. Memori yang digunakan adalah *microSD* yang telah terpasang pada modul *microSD Adapter*. Perancangan konfigurasi antara *pin* arduino mega dengan memori eksternal dapat dilihat pada Gambar 3.17 *Wiring Modul microSD Adapter* dengan Arduino Mega. Berikut adalah sambungan antara *microSD Adapter* dengan Arduino Mega :

1. CS (*Chip Select*) – Pin 53 (SS) Arduino Mega
2. SCK (*Serial Clock*) – Pin 52 (SCK) Arduino Mega
3. MISO (*Serial Data Out*) – Pin 50 (MISO) Arduino Mega
4. MOSI (*Serial Data In*) – Pin 51 (MOSI) Arduino Mega
5. VCC – Pin 5 Volt Arduino Mega
6. GND – Pin *Ground* Arduino Mega



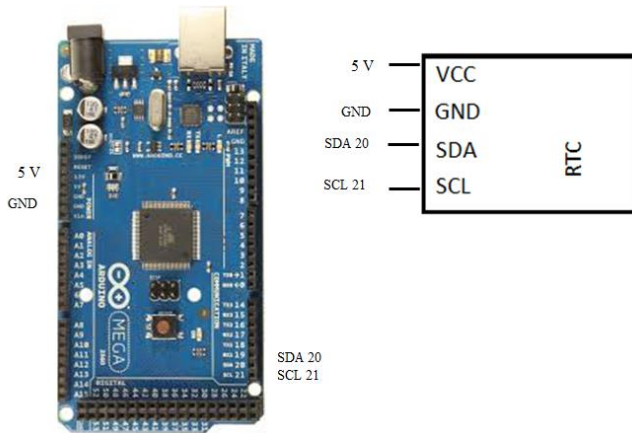
Gambar 3.17 Wiring Modul *Micro SD Adapter* dengan Arduino Mega

3.2.4.8 Perancangan *Real Time Clock (RTC)*

Untuk dapat mencatat kondisi secara *real time* dibutuhkan penghitungan waktu yang akurat. Penghitung ini harus dapat menghitung jam, menit, detik, tanggal, bulan dan tahun secara akurat. Untuk penghitungan waktu tersebut digunakan modul RTC DS1307. Dimana modul ini sudah memiliki baterai sebagai sumber ketika sumber utamanya habis atau tidak ada. Sehingga meski sumber terputus, penghitung waktu akan tetap berjalan.

Modul RTC DS1307 dapat langsung dihubungkan pada arduino. Perancangan konfigurasi antara *pin* arduino mega dengan RTC dapat dilihat pada Gambar 3.18 *Wiring* Modul RTC DS1307 dengan Arduino Mega. Sambungan dengan Arduino Mega dapat dituliskan sebagai berikut :

1. VCC – pin 5V Arduino Mega
2. GND – pin *Ground* Arduino Mega
3. SDA – pin 20 (SDA) pada Arduino Mega
4. SCL – pin 21 (SDL) pada Arduino Mega



Gambar 3.18 Wiring Modul RTC DS1307 dengan Arduino Mega

3.3 Perancangan *Software*

Pada perancangan *software* pada tugas akhir ini yang dibahas meliputi pemrograman arduino IDE pada arduino Uno untuk *Slave* serta arduino mega untuk *Master*.

Perancangan *software* dalam pembuatan alat yang digunakan oleh Biner Rochmatul C.A yakni pada Sub Bab 3.3.1 Perancangan *Software Slave*.

Perancangan *software* dalam pembuatan alat yang digunakan oleh M. Fatkhur Rozi yakni pada Sub Bab 3.3.2 Perancangan *Software Master*.

Perancangan *software* dalam pembuatan alat yang digunakan oleh Biner Rochmatul C.A yakni :

3.3.1 Perancangan *Software Slave*

Merupakan perancangan *software* yang digunakan dalam membantu pencatatan kWh yang digunakan oleh pelanggan. Perancangan *Software Slave* meliputi perancangan *flowchart* dan perancangan pemrograman. Dalam perancangan *software Slave* juga terdiri dari beberapa pemrograman, diantaranya pemrograman LCD, LDR, PLC, dan Arduino Uno.

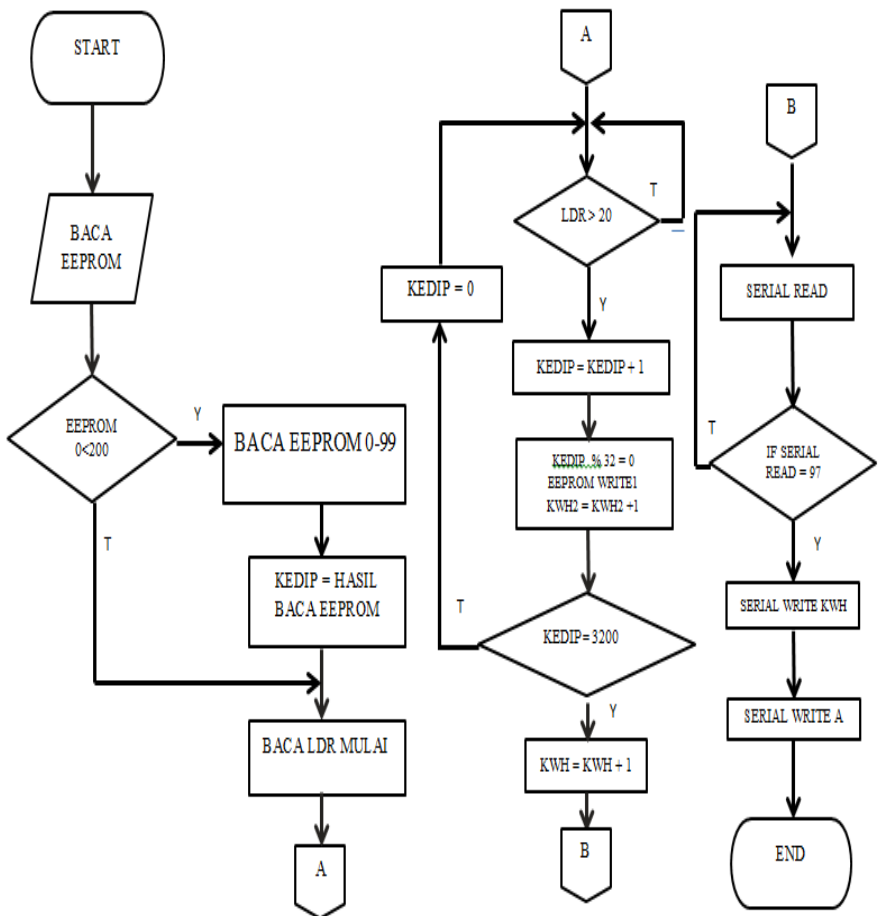
Perancangan *flowchart* membantu dalam pembuatan perancangan program *Slave* yang akan dibuat. Perancangan *flowchart*

Slave dapat dilihat pada Gambar 3.19 *Flowchart* perancangan *software Slave*

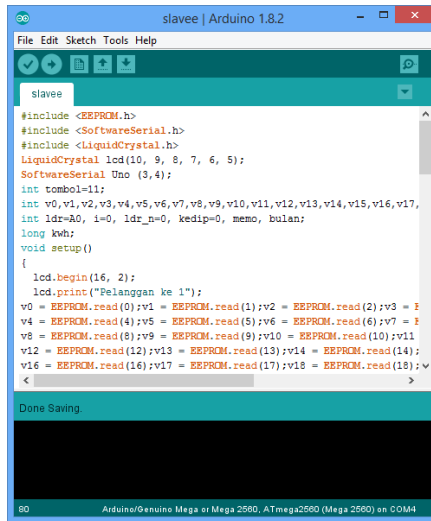
Perancangan *flowchart* membantu dalam pembuatan perancangan pemrograman *Slave* yang akan dibuat. Algoritma dari perancangan pemrograman *Slave* dari *flowchart* perancangan *software Slave* yakni:

1. Pembacaan memori EEPROM pada arduino yang tersimpan secara sementara.
2. Ketika *impuls* berkedip dengan nilai lebih besar sama dengan 20 dengan delay 0,1 s, maka nilai tersebut akan disimpan pada EEPROM.
3. Nilai LDR yang lebih dari sama dengan 20 tersebut maka nilai kedip akan bertambah 1. Pembacaan algoritma ini diulang-ulang dari langkah pertama hingga nilai kedip sama dengan 3200.
4. Jika nilai kedip *modulus* 32 bernilai sama dengan 0, maka nilai kedip tersebut akan disimpan pada EEPROM
5. Ketika nilai kedip telah bernilai sama dengan 3200 maka nilai kWh akan bertambah 1 dan disimpan pada EEPROM
6. Nilai kWh yang terus bertambah tersebut akan ditampilkan pada LCD 16 x 2.
7. Nilai kWh tersebut akan dikirim kepada *hardware Master*, jika *Slave* telah menerima perintah mengirim dari *Master*.

Perancangan pemrograman membantu optimasi perancangan *hardware* guna mencatat kWh yang digunakan oleh pelanggan. Perancangan pemrograman ini dapat dilihat pada Gambar 3.20 Perancangan Pemrograman *Slave*. Berikut merupakan program yang digunakan dalam perancangan pemrograman *Slave* :



Gambar 3.19 Flowchart Perancangan Software Slave



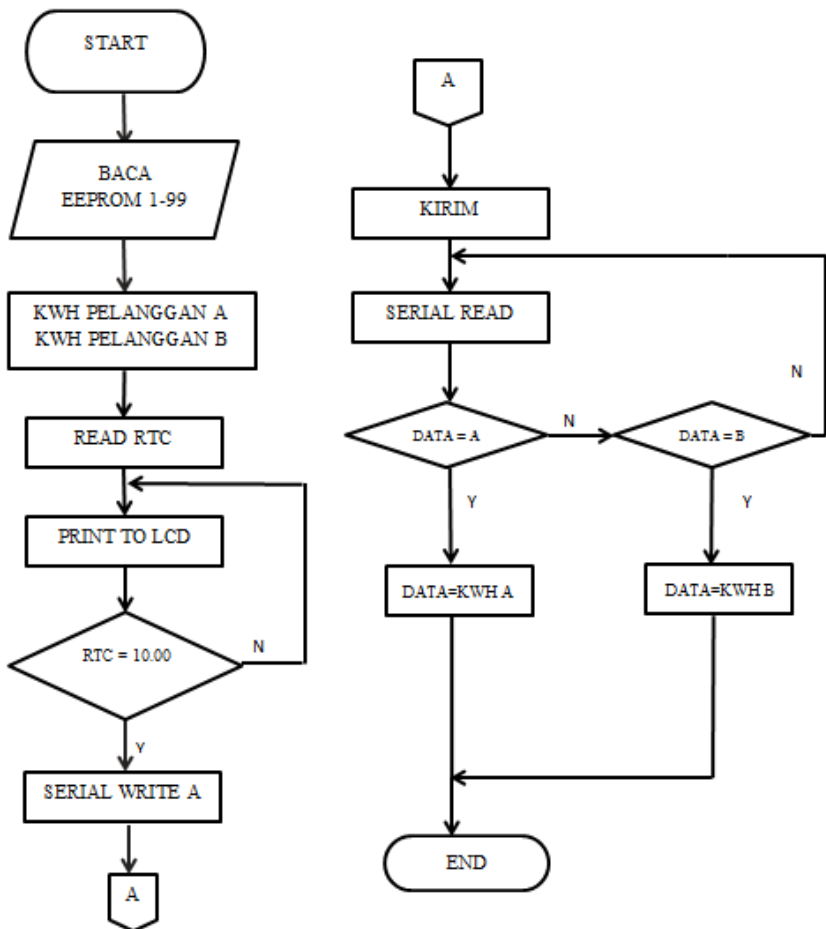
Gambar 3.20 Perancangan Pemrograman *Slave*

Perancangan *software* dalam pembuatan alat yang digunakan oleh M. Fatkhur Rozi yakni :

3.3.2 Perancangan *Software Master*

Merupakan perancangan *software* yang digunakan dalam membantu menghimpun pencatatan kWh yang digunakan oleh pelanggan pada panel PHB. Perancangan *Software Master* meliputi perancangan *flowchart* dan perancangan pemrograman. Dalam perancangan *software Master* juga terdiri dari beberapa pemrograman, diantaranya pemrograman LCD, LDR, PLC, RTC, Memori, dan Arduino Mega. Perancangan *flowchart Master* dapat dilihat pada Gambar 3.21 *Flowchart perancangan software Master*

Perancangan *flowchart* membantu dalam pembuatan perancangan program *Master* yang akan dibuat. Perancangan pemrograman ini dapat dilihat pada Gambar 3.22 Perancangan Pemrograman *Master*. Perancangan Pemrograman *Master* Berikut merupakan perancangan *flowchart Master* :

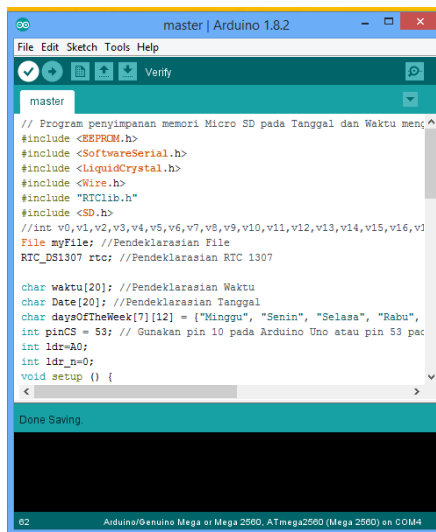


Gambar 3.21 Flowchart Perancangan Software Master

Perancangan *flowchart* membantu dalam pembuatan perancangan pemrograman *Slave* yang akan dibuat. Algoritma dari perancangan pemrograman *Slave* dari *flowchart* perancangan *software Slave* yakni:

1. Pembagian pembacaan memori EEPROM pada arduino mega menjadi delapan bagian.

2. Pembacaan pada RTC mengenai waktu kemudian ditampilkan terhadap LCD 20 x 4.
3. *Master* akan mengirimkan perintah kepada *Slave* untuk mengirimkan data kWh secara berkala dengan waktu yang telah ditentukan yakni pukul 10.00.
4. Pembacaan jenis data kWh yang ada, jika data yang terbaca sama dengan Data A, maka termasuk dalam nilai kWh A bertambah.
5. Hal itu juga berlaku bila data yang terbaca sama dengan Data B, mak termasuk dalam nilai kWh B bertambah.



BAB IV

PENGUJIAN DAN ANALISA DATA

Untuk mengetahui hasil dari pembuatan alat dalam kegiatan Tugas Akhir ini, maka perlu dilakukan pengujian serta menganalisa data yang diperoleh. Pada bab ini akan dibahas mengenai hasil pengujian serta analisa data yang diperoleh. Dari *hardware* yang telah dibuat tersebut, selanjutnya Biner Rochmatul C.A akan melakukan pengujian dan analisa dari berbagai komponen sistem diantaranya pada Sub Bab 4.1 Pengujian *Slave*.

Dari *hardware* yang telah dibuat tersebut, selanjutnya M. Fatkhur Rozi akan melakukan pengujian dan analisa dari berbagai komponen sistem diantaranya pada Sub Bab 4.2 Pengujian *Master*.

Dari *hardware* yang telah dibuat tersebut, selanjutnya Biner Rochmatul C.A dan M. Fatkhur Rozi akan melakukan pengujian dan analisa dari berbagai komponen sistem diantaranya pada Sub Bab 4.3 Pengujian KWh Meter dan 4.4 Pengujian *Prototype* Simulator (Sistem Keseluruhan).

Dari *hardware* yang telah dibuat tersebut, selanjutnya Biner Rochmatul C.A akan melakukan pengujian dan analisa dari berbagai komponen sistem diantaranya sebagai berikut :

4.1 Pengujian *Slave*

Pengujian *Slave* merupakan pengujian secara serentak terhadap *Power Line Carrier*, LCD, dan LDR. Proses pengujian terhadap *Slave* yakni dengan cara pengujian terhadap pengukuran kWh meter dengan menggunakan perhitungan *impuls*. Impuls yang dihasilkan oleh KWh meter dideteksi menggunakan sensor LDR. *Impuls* tersebut lalu dikonversikan terhadap satuan kWh. Proses pengkonversian dari satuan *impuls* menjadi satuan kWh melalui pemrograman terhadap arduino uno yang terdapat pada *Slave*. Hasil pengukuran tersebut lalu akan ditampilkan pada LCD 16 x 2. Baik untuk pengukuran nilai kWh pada pelanggan A maupun pada pelanggan B. Selain data pengukuran KWh tersebut ditampilkan pada LCD juga diproses dengan pemrograman pada arduino uno untuk dikirimkan ke *Master* yang terdapat pada *prototype* panel PHB setiap harinya melalui media *Power Line Carrier*. Program pengujian untuk *Slave* dapat dilihat pada Lampiran A.1 Pemrograman *Slave*. Pengujian ini dapat dilihat pada

Gambar 4.1 Pengujian *Slave*.

Pada Tabel 4.1 Pengujian *Impuls* dilakukan dengan cara menghubungkan beban resistif berupa lampu dengan nilai daya yang beragam dengan KWh Meter. KWh Meter akan menghasilkan *impuls* yang terbaca dari beban yang dihubungkan dengan KWh Meter. *Impuls* yang timbul kemudian dihitung dalam rentang waktu 1 menit. Pada Tabel 4.1 Pengujian *Impuls* ini menghasilkan nilai *impuls* yang sangat kecil. Hal ini juga karena dipengaruhi oleh beban yang digunakan. Jika pembebanan yang digunakan nilainya cukup besar, maka terlihat secara jelas pertambahan nilai kWh yang terhitung. Maka untuk mengetahui nilai kWh dengan pembebanan yang kecil dan waktu yang singkat didapat dengan Persamaan 4.1 yakni sebagai berikut :

$$\text{Nilai kW} = \frac{n \times 3600}{\text{Konstanta} \times t} \quad (4.1)$$

Dimana n adalah nilai *impuls* yang terbaca oleh *Slave*, t adalah waktu pengambilan data, dan *Konstanta* yakni nilai satuan *impuls* / kWh yang terdapat pada *nameplate* KWh meter sehingga kami menggunakan nilai 3200 karena nilai *impuls* / kWh yang terdapat pada *nameplate* yakni 3200 *impuls* / kWh.

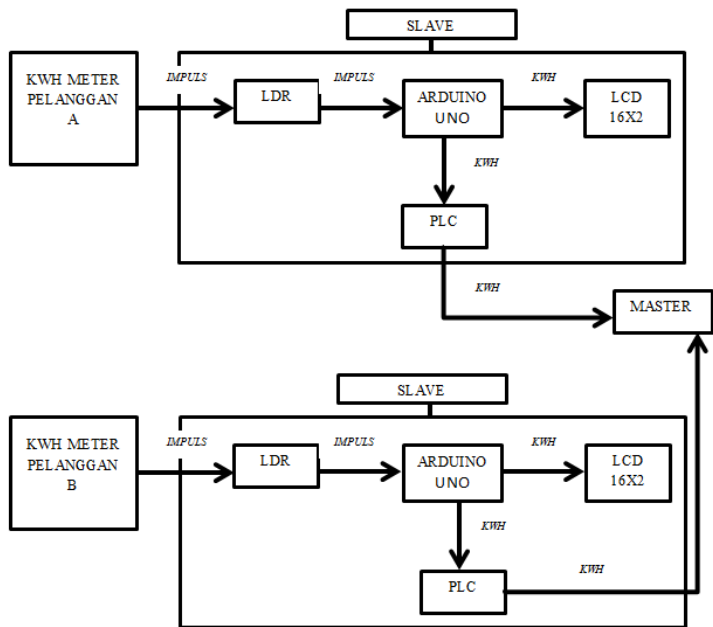
Tabel 4.1 Pengujian *Impuls*

No	Beban (W)	<i>Impuls</i> / Menit	<i>Impuls</i> yang Terbaca oleh <i>Slave</i>	Nilai kW
1	25	1	1	0,02
2	50	2	2	0,04
3	75	3	3	0,06
4	100	4	4	0,08
5	125	5	5	0,09
6	150	6	6	0,11

Pada Tabel 4.2 Pengujian *Slave* telah ditunjukkan kWh yang terukur dan yang telah terkirim pada *Master*, baik untuk pelanggan A maupun pelanggan B. Pada sisi kWh Terukur dan kWh Terkirim tidak terjadi perbedaan yang mencolok. Hal ini dikarenakan jaringan yang digunakan oleh media *Power Line Carrier* dalam keadaan normal. Jarak pengujian PLC yang digunakan yakni sejauh 5 m. Pembebanan yang digunakan yakni berupa bohlam lampu yang memiliki nilai daya yang beragam yang bersifat resistif. Tidak adanya gangguan pembebanan yang mempengaruhi pengiriman data kWh.

Tabel 4.2 Pengujian *Slave*

No	Beban (W)	Jarak PLC (m)	Pelanggan A		Pelanggan B	
			kWh Terukur	kWh Terkirim	kWh Terukur	kWh Terkirim
1	25	5	72	72	72	72
2	50	5	144	144	144	144
3	75	5	216	216	216	216
4	100	5	288	288	288	288
5	125	5	324	324	324	324
6	150	5	396	396	396	396



Gambar 4.1 Pengujian *Slave*

Pada *Slave* ini terdiri dari beberapa komponen meliputi *Light Dependent Resistor* (LDR), LCD, *Power Line Carrier* (PLC), LM3805, dan *power supply*. Dari beberapa komponen tersebut maka dilakukan beberapa pengujian, diantaranya sebagai berikut :

4.1.1 Pengujian *Power Line Carrier* (PLC)

Pengujian *Power Line Carrier* (PLC) merupakan salah satu pengujian komponen yang terdapat pada *hardware Slave*. *Power Line Carrier* merupakan media komunikasi data antara *Master* dengan *Slave*. Media ini berfungsi sebagai penyalur data pengukuran kWh yang telah dilakukan oleh *Slave* terhadap kWh Meter pelanggan.

Proses pengujian pada *Power Line Carrier* ini adalah dengan menggunakan dua modul yakni modul *Slave* dan modul *Master*. Proses pengujian ini dapat dilihat pada Gambar 4.2 Pengujian Jarak *Power Line Carrier*. Masing-masing modul dihubungkan dengan jalur fasa PLN yang sama pada sisi Fasa modul *Slave* dan *Master* dan dihubungkan dengan netral PLN pada sisi netral modul *Slave* dan *Master*. Setelah sisi fasa dan netral terhubung pada sisi fasa dan netral PLN kemudian ditentukan jarak pengujian data yang akan dikirim. Kemudian data yang akan diuji untuk pengiriman diupload pada arduino uno yang terdapat pada *Slave* untuk dikirim pada arduino mega pada *Master*. Program pengujian untuk *power line carrier* dapat dilihat pada Lampiran A.1 Pemrograman *Slave* untuk pada sisi modul *Slave* dan Lampiran A.4 Pemrograman *Master* untuk pada sisi modul *Master*. Pada proses pengujian ini juga dilaksanakan di area Perpustakaan ITS dengan menggunakan satu jalur fasa. Hasil pada proses pengujian ini ditunjukkan pada Tabel 4.3 Pengujian Jarak *Power Line Carrier*.

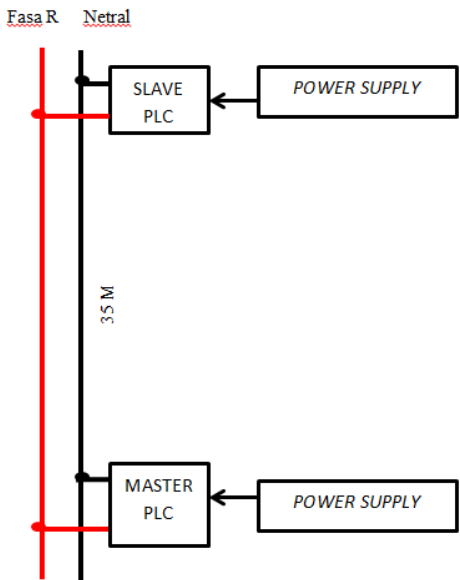
Pada Tabel 4.3 Pengujian Jarak *Power Line Carrier* ini juga mengambil data dari 4 jarak, yakni 5 m, 17,5 m, 27,5 m, dan 35 m. Perhitungan jarak pengujian dimulai dari letak *Slave* PLC sampai dengan letak *Master* PLC. Pada masing-masing jarak dilakukan pengiriman data dari angka 1 hingga 1000. Perbandingan data yang terkirim pada *Slave* dan diterima oleh *Master* inilah yang telah kami amati. Pada *Slave* digunakan pengiriman data “24” kemudian data yang diterima pada *Master* yakni “0024” pada jarak 5 m. Pada *Slave* digunakan pengiriman data “16” kemudian data yang diterima pada *Master* yakni “0016” pada jarak 17,5 m. Pada *Slave* digunakan pengiriman data “48” kemudian data yang diterima pada *Master* yakni “0048” pada jarak 27,5 m. Pada *Slave* digunakan pengiriman data “33” kemudian data yang diterima pada *Master* yakni “0033” pada jarak 35 m.

Proses komunikasi pada PLC tidak akan mengalami gangguan selama pada proses pengiriman data yang terdapat pada jaringan listrik dalam satu fasa tersebut pembebanannya normal yakni hanya terdapat beban resistif saja. Dan waktu pengiriman data terdapat *delay*. Sehingga

tidak terjadi pengiriman data secara terus menerus. Apabila terjadi pengiriman data secara terus menerus akan mengakibatkan penumpukan data antara data sebelumnya dengan setelahnya.

Tabel 4.3 Pengujian Jarak *Power Line Carrier*

No	Jarak (M)	Slave	Master
1.	5	24	0024
2.	17,5	16	0016
3.	27,5	48	0048
4.	35	33	0033



Gambar 4.2 Pengujian Jarak *Power Line Carrier*

4.1.2 Pengujian LCD

Pengujian LCD 16 x 2 merupakan salah satu pengujian komponen yang terdapat pada *hardware Slave*. LCD ini berfungsi untuk menampilkan data hasil pengukuran KWh Meter yang terdeteksi oleh LDR yang kemudian diproses oleh arduino Uno untuk ditampilkan di LCD.

Proses pengujian LCD ini dengan mengecek *wiring* LCD dengan Arduino Uno. Berikut merupakan sambungan antara Arduino Uno

:

1. Katoda – Pin GND Arduino Uno
2. Anoda – Pin 5 V Arduino Uno
3. D7 – Pin 5 Arduino Uno
4. D6 – Pin 6 Arduino Uno
5. D5 – Pin 7 Arduino Uno
6. D4 – Pin 8 Arduino Uno
7. E – Pin 9 Arduino Uno
8. RW – Pin GND Arduino Uno
9. RS – Pin 10 Arduino Uno
10. V0 – Keluaran Potensio
11. VDD – Pin 5V Arduino Uno
12. VSS – Pin GND Arduino Uno

Setelah dilakukan pengecekan maka langkah selanjutnya adalah dengan menggunakan program “Hello World!” guna memastikan apakah LCD 16 x 2 dapat berfungsi dengan baik ataupun tidak. Karena apabila terjadi kesalahan maka tulisan “Hello World!” tidak akan muncul dan LCD tidak akan berfungsi. Program pengujian untuk LCD 16 x 2 dapat dilihat pada Lampiran A.2 Pemrograman LCD 16 x 2.



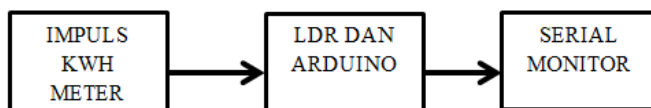
Gambar 4.3 Pengujian LCD

4.1.3 Pengujian *Light Dependent Resistor* (LDR)

Pengujian *Light Dependent Resistor* (LDR) ini dilakukan dengan cara meletakkan sensor LDR tepat pada depan indikator *impuls* pada kWh meter. Pengujian ini dapat dilihat pada Gambar 4.4 Pengujian LDR. Sensor LDR ini terhubung dengan Arduino dan laptop. Sehingga nilai LDR ini dapat dilihat dari Serial Monitor yang terdapat pada tampilan Arduino pada laptop baik ketika impuls pada kWh meter dalam keadaan berkedip maupun padam. Program pengujian untuk *Light Dependent Resistor* dapat dilihat pada Lampiran A.3 Pemrograman *Light Dependent Resistor*. Ketika *impuls* berkedip maka LDR akan mendeteksi dan nilai *impuls* akan bertambah 1. Ketika *impuls* berkedip maka nilai LDR akan meningkat. Nilai LDR akan bernilai antara 22 hingga 30 ketika *impuls* berkedip. Namun ketika indikator *impuls* tidak berkedip maka nilai LDR akan bernilai kurang dari sama dengan 1. Hasil pengujian ini dapat dilihat pada Tabel 4.4 ketika impuls dalam keadaan berkedip nilai LDR yang ditunjukkan pada serial monitor yakni memiliki rentang nilai dari 23 hingga 27. Sedangkan ketika impuls dalam keadaan padam maka nilai LDR pada serial monitor akan bernilai 1.

Tabel 4.4 Pengujian *Light Dependent Resistor* (LDR)

No	Keadaan Impuls KWh	Nilai LDR
1.	Berkedip	23
		23
		25
		26
		26
		27
		27
2.	Padam	1
		1
		1
		1
		1
		1
		1



Gambar 4.4 Pengujian LDR

Dari *hardware* yang telah dibuat tersebut, selanjutnya M. Fatkhur Rozi akan melakukan pengujian dan analisa dari berbagai komponen sistem diantaranya sebagai berikut :

4.2 Pengujian Master

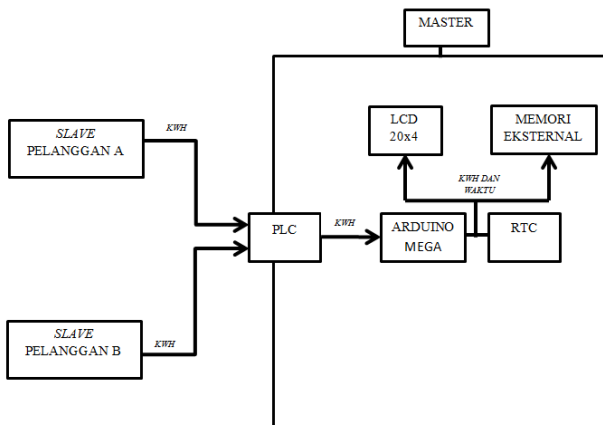
Pengujian *Master* merupakan pengujian secara serentak terhadap *Power Line Carrier*, LCD, RTC, dan Memori. Pengujian *Master* ini dilakukan dengan cara menerima data kWh yang telah dikirim oleh *Slave*, baik data kWh pelanggan A maupun data kWh Pelanggan B. Pengiriman data oleh *Slave* kepada *Master* ini melalui media *Power Line Carrier*. *Power Line Carrier* berfungsi sebagai media pengiriman serta penerimaan data antara *Slave* dan *Master*. Data kWh yang dikirimkan secara berkala oleh *Slave* tersebut kemudian disimpan oleh EEPROM yang kemudian untuk ditampilkan pada LCD 20 x 4 beserta dengan waktu penerimaannya. LCD 20 X 4 disini berfungsi untuk menampilkan data yang telah dikirim oleh *Slave* terhadap *Master*. LCD untuk melihat data kWh Pelanggan A ataupun data kWh Pelanggan B yang tercatat beserta waktu pencatatannya. Dengan menekan *Push Button* juga dapat mengalihkan tampilan LCD menjadi nemampilkan selisih kWh antara kWh yang digunakan oleh pelanggan pada bulan kemarin dan juga pemakaia kWh bulan ini. *Real Time Clock* (RTC) pada *Master* berfungsi sebagai pencatatan waktu pengiriman data kWh yang kemudian ditampilkan oleh LCD juga akan disimpan pada memori kartu (SD Card). Data kWh yang telah dikirim oleh *Slave* secara berkala pada waktu-waktu tertentu tersebut kemudian disimpan pada memori berbentuk kartu yang telah disediakan pada *Master* guna mempermudah pencatatan penggunaan nilai kWh yang ada pada pelanggan. Pengujian ini dapat dilihat pada Gambar 4.5 Pengujian *Master*. Program pengujian untuk *Master* dapat dilihat pada Lampiran A.4 Pemrograman *Master*.

Pada Tabel 4.5 Pengujian *Master* dapat dilihat data penggunaan kW oleh pelanggan A dan data penggunaa kW oleh pelanggan B yang telah dikirim oleh *Slave* serta waktu pencatatan pengiriman data kWh tersebut. Data kWh yang terkirim pada *Master* tidak terdapat perbedaan yang mencolok dengan data kWh yang terdapat pada *Slave*. Hal ini dikarenakan jaringan yang digunakan oleh media *Power Line Carrier* (PLC) dalam keadaan normal. Jarak pengujian PLC yang digunakan yakni sejauh 5 m. Pembebanan yang digunakan yakni berupa bohlam lampu yang memiliki nilai daya yang beragam yang bersifat resistif. Tidak adanya gangguan pembebanan yang mempengaruhi pengiriman

data kWh. Waktu pengiriman memiliki tetapan pada pukul 10.00 WIB pagi. Hal ini disesuaikan dengan waktu pencatatan petugas pencatat meter PLN. Namun ada selisih waktu antara pengiriman data pengukuran kWh antara pelanggan A dengan pelanggan B sebesar 2 detik.

Tabel 4.5 Pengujian *Master*

No	Jarak PLC (m)	Pelanggan A (KWh Terkirim)	Pelanggan B (KWh Terkirim)	Memori Pelanggan A	Memori Pelanggan B	Waktu Pelanggan A	Waktu Pelanggan B
1	5	72	72	72	72	10.00.00	10.00.02
2	5	144	144	144	144	10.00.00	10.00.02
3	5	216	216	216	216	10.00.00	10.00.02
4	5	288	288	288	288	10.00.00	10.00.02
5	5	324	324	324	324	10.00.00	10.00.02
6	5	396	396	396	396	10.00.00	10.00.02



Gambar 4.5 Pengujian *Master*

Pada *Master* ini terdiri dari beberapa komponen meliputi *Light Dependent Resistor* (LDR), LCD, *Power Line Carrier* (PLC), LM3805, Memori, RTC, dan *power supply*. Dari beberapa komponen tersebut maka dilakukan beberapa pengujian, diantaranya sebagai berikut :

4.2.1 Pengujian *Power Line Carrier* (PLC)

Pengujian *Power Line Carrier* (PLC) merupakan salah satu pengujian komponen yang terdapat pada *hardware Master*. *Power Line Carrier* merupakan media komunikasi data antara *Master* dengan *Slave*. Media ini berfungsi sebagai penyalur data pengukuran kWh yang telah dilakukan oleh *Slave* terhadap KWh Meter.

Proses pengujian pada *Power Line Carrier* ini adalah dengan menggunakan dua modul yakni modul *Slave* dan modul *Master*. Proses pengujian ini dapat dilihat pada Gambar 4.2 Pengujian Jarak *Power Line Carrier*. Masing-masing modul dihubungkan dengan jalur fasa PLN yang sama pada sisi Fasa modul *Slave* dan *Master* dan dihubungkan dengan netral PLN pada sisi netral modul *Slave* dan *Master*. Setelah sisi fasa dan netral terhubung pada sisi fasa dan netral PLN kemudian ditentukan jarak pengujian data yang akan dikirim. Kemudian data yang akan diuji untuk pengiriman diupload pada arduino uno yang terdapat pada *Slave* untuk dikirim pada arduino mega pada *Master*. Program pengujian untuk *power line carrier* dapat dilihat pada Lampiran A.1 Pemrograman *Slave* untuk pada sisi modul *Slave* dan Lampiran A.4 Pemrograman *Master* untuk pada sisi modul *Master*. Pada proses pengujian ini juga dilaksanakan di area Perpustakaan ITS dengan menggunakan satu jalur fasa. Hasil pada proses pengujian ini ditunjukkan pada Tabel 4.6 Pengujian Jarak *Power Line Carrier*.

Pada Tabel 4.6 Pengujian Jarak *Power Line Carrier* ini juga mengambil data dari 4 jarak, yakni 5 m, 17,5 m, 27,5 m, dan 35 m. Perhitungan jarak pengujian dimulai dari letak *Slave* PLC sampai dengan letak *Master* PLC. Pada masing-masing jarak dilakukan pengiriman data dari angka 1 hingga 1000. Perbandingan data yang terkirim pada *Slave* dan diterima oleh *Master* inilah yang telah kami amati. Pada *Slave* digunakan pengiriman data “44” kemudian data yang diterima pada *Master* yakni “0044” pada jarak 5 m. Pada *Slave* digunakan pengiriman data “29” kemudian data yang diterima pada *Master* yakni “0029” pada jarak 17,5 m. Pada *Slave* digunakan pengiriman data “48” kemudian data yang diterima pada *Master* yakni “0048” pada jarak 27,5 m. Pada *Slave* digunakan pengiriman data “33” kemudian data yang diterima pada *Master* yakni “0019” pada jarak 35 m.

Proses komunikasi pada PLC tidak akan mengalami gangguan selama pada proses pengiriman data yang terdapat pada jaringan listrik dalam satu fasa tersebut pembebanannya normal. Pembebanan pada saat pengujian menggunakan pembebanan resistif berupa bohlam lampu. Dan waktu pengiriman data terdapat *delay*. Sehingga tidak terjadi pengiriman

data secara terus menerus. Apabila terjadi pengiriman data secara terus menerus akan mengakibatkan penumpukan data antara data sebelumnya dengan setelahnya.

Tabel 4.6 Pengujian Jarak *Power Line Carrier*

No	Jarak (m)	<i>Slave</i>	<i>Master</i>
1.	5	44	0044
2.	17,5	29	0029
3.	27,5	20	0020
4.	35	19	0019

4.2.2 Pengujian LCD

Pengujian LCD 20 x 4 merupakan salah satu pengujian komponen yang terdapat pada *hardware Master*. LCD ini berfungsi untuk menampilkan data hasil pengukuran KWh Meter yang dilakukan oleh *Slave* kemudian dihimpun oleh *Master*.

Proses pengujian LCD ini dengan mengecek *wiring* LCD dengan Arduino Mega. Berikut merupakan sambungan antara Arduino Mega :

1. SCL – Pin A5 Arduino Mega
2. SDA – Pin A4 Arduino Mega
3. VCC – Pin 5 Volt Arduino Mega
4. GND – Pin *Ground* Arduino Mega

Setelah dilakukan pengecekan maka langkah selanjutnya adalah dengan menggunakan program menampilkan hasil pengukuran kWh pelanggan guna memastikan apakah LCD 20 x 4 dapat berfungsi dengan baik ataupun tidak. Karena apabila terjadi kesalahan maka hasil pengukuran kWh pelanggan tidak akan akan muncul dan LCD tidak akan berfungsi. Program pengujian untuk LCD 20 x 4 dapat dilihat pada Lampiran A.5 Pemrograman LCD 20 x 4. Hasil pengujian LCD ini dapat dilihat pada Gambar 4.6 Pengujian LCD 20x4.



Gambar 4.6 Pengujian LCD 20 x 4

4.2.3 Pengujian Memori Eksternal

Untuk melakukan pengujian pada modul *microSD Adapter*, pertama-tama dapat langsung dihubungkan pada arduino. Berikut adalah sambungan antara *microSD Adapter* dengan Arduino Mega :

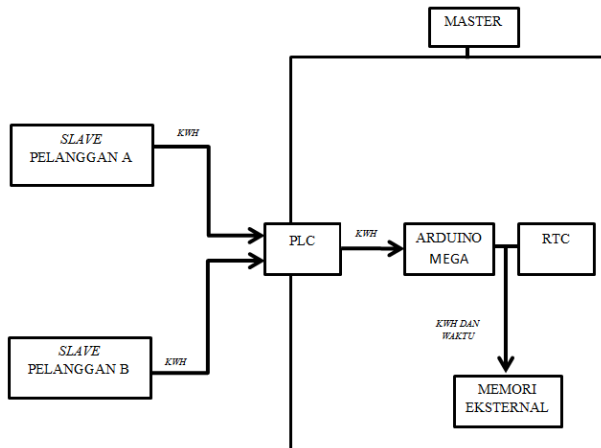
1. CS (*Chip Select*) – Pin 53 (SS) Arduino Mega
2. SCK (*Serial Clock*) – Pin 52 (SCK) Arduino Mega
3. MISO (*Serial Data Out*) – Pin 50 (MISO) Arduino Mega
4. MOSI (*Serial Data In*) – Pin 51 (MOSI) Arduino Mega
5. VCC – Pin 5 Volt Arduino Mega
6. GND – Pin *Ground* Arduino Mega

Setelah dilakukan pengecekan sambungan antara arduino mega dengan modul *microSD Adapter* maka dilakukan pemrograman pada *microSD Adapter*. Program pengujian untuk memori eksternal dapat dilihat pada Lampiran A.6 Pemrograman Memori Eksternal. *MicroSD Adapter* akan menyimpan setiap data kWh yang dikirim oleh *Slave* baik pada Pelanggan A maupun Pelanggan B. Proses pengujian ini dapat dilihat pada Gambar 4.7 Pengujian Memori Eksternal. Pengujian memori ini dilakukan dengan mekanisme penyimpanan data kWh yang telah dikirim pada *Master* seperti yang dapat dilihat pada Tabel 4.7 Pengujian Memori Eksternal. Data yang tersimpan tersebut akan disimpan sesuai dengan waktu pengirimannya. Sehingga pada pencatatan penyimpanan data kWh pelanggan, juga disertai dengan pencatatan waktu pengiriman data. Penguji dapat mengambil data pada memori dengan cara melepas *microSD Adapter* kemudian mendownload data tersebut menggunakan laptop. Data yang tersimpan pada *microSD Adapter* hanya dapat dilihat dengan menggunakan laptop.

Tabel 4.7 Pengujian Memori Eksternal

No	Data Yang Dikirim	Data Pada Memori
1.	72	72
2.	144	144
3.	216	216
4.	288	288
5.	324	324

6.	396	396
----	-----	-----



Gambar 4.7 Pengujian Memori Eksternal

4.2.4 Pengujian *Real Time Clock* (RTC)

Proses pengujian modul RTC DS1307 dapat langsung dihubungkan pada arduino. Sambungan dengan Arduino Mega dapat dilakukan pengecekan terlebih dahulu, yakni sebagai berikut :

1. VCC – pin 5V Arduino Mega
2. GND – pin *Ground* Arduino Mega
3. SDA – pin 20 (SDA) pada Arduino Mega
4. SCL – pin 21 (SDL) pada Arduino Mega

Setelah dilakukan pengecekan sambungan antara arduino mega dengan modul RTC DS1307 maka dilakukan pemrograman pada RTC yang juga dihubungkan dengan LCD yang berguna untuk mempermudah dalam pengujian RTC ini. Setelah dilakukan pemrograman, pada LCD akan menampilkan waktu sesuai dengan RTC yang menunjukkan waktu secara akurat yang kemudian kami bandingkan dengan waktu yang ditunjukkan oleh laptop kami. Proses pengujian ini dapat dilihat pada Gambar 4.8 Pengujian *Real Time Clock* (RTC). Program pengujian untuk RTC dapat dilihat pada Lampiran A.7 Pemrograman *Real Time Clock*.

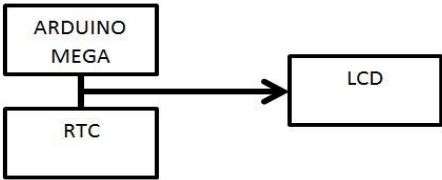
Pada Tabel 4.8 Pengujian *Real Time Clock* (RTC) jika dibandingkan waktu yang ditunjukkan pada laptop dengan waktu yang

ditampilkan oleh RTC dengan bantuan LCD tidak mengalami selisih waktu yang cukup drastis, bahkan cenderung sama. Pada LCD menunjukkan waktu pukul 10.00 dan pada laptop juga menunjukkan waktu 10.00. Penunjukkan waktu oleh modul RTC DS1307 dengan laptop tidak akan mengalami selisih apabila waktu penguploadan program memiliki waktu yang tepat. Apabila terdapat selisih waktu antara waktu yang ditunjukkan oleh laptop dengan waktu yang ditunjukkan oleh modul RTC DS1307 yang kemudian ditampilkan oleh LCD 20 x 4 maka proses penguploadan tersebut tidak tepat.

Real Time Clock memiliki fungsi untuk menyimpan informasi detik, menit, jam, hari, tanggal, bulan, dan tahun. Disebut sebagai *Real Time* karena tanggal pada akhir bulan secara otomatis disesuaikan dengan bulan dengan kurang dari 31 hari, termasuk koreksi untuk tahun kabisat. Jam beroperasi baik dalam 24 jam atau format 12 jam dengan indikator AM/PM aktif ketika kondisi *low*.

Tabel 4.8 Pengujian *Real Time Clock* (RTC)

No	Waktu pada RTC	Waktu pada Laptop	Selisih Waktu (Menit)
1.	10.00	10.00	0
2.	10.10	10.10	0
3.	10.20	10.20	0
4.	10.30	10.30	0
5.	10.40	10.40	0



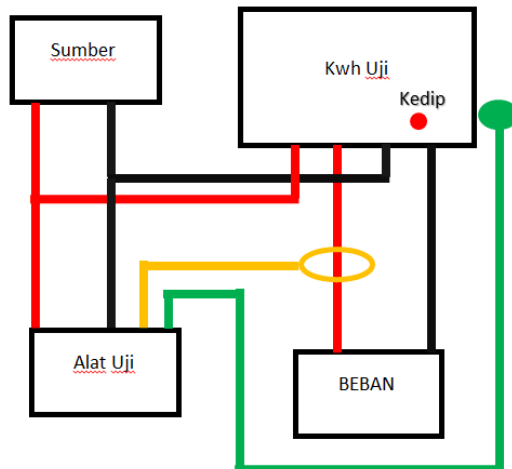
Gambar 4.8 Pengujian *Real Time Clock* (RTC)

Dari *hardware* yang telah dibuat tersebut, selanjutnya Biner Rochmatul C.A dan M. Fatkhur Rozi akan melakukan pengujian dan analisa dari berbagai komponen sistem diantaranya sebagai berikut :

4.3 Pengujian KWh Meter

Proses pengujian KWh Meter ini dilakukan dengan cara

menghubungkan KWh meter dengan Alat Uji. Alat uji ini terdiri dari sensor arus yang berupa tang ampere di rangkai parallel dengan kWh meter dan juga sensor arus yang dipasang di salah satu *line* serta sensor kedip kWh yang di pasang pada indikator kedip *impuls* kWh. Seperti yang dapat dilihat pada Gambar 4.9 Pengujian Tingkat Kesalahan KWh Meter. Sensor kedip KWh digambarkan dengan garis berwarna hijau. Sensor arus yang berupa tang ampere digambarkan dengan garis berwarna kuning. Pemasangan pada sumber, alat uji, KWh Meter yang akan diuji, dan beban dilakukan secara paralel dengan digambarkan garis warna merah untuk sisi fasa dan garis hitam untuk sisi netralnya.



Gambar 4.9 Pengujian Tingkat Kesalahan KWh Meter

Pada pengujian KWh Meter ini diperoleh tingkat kesalahan atau prosentase *error* dari masing-masing KWh Meter yang digunakan. Tingkat ketelitian atau prosentase *error* dari masing-masing KWh Meter dapat dilihat pada Tabel 4.9 Pengujian Ketelitian KWh Meter *Master*, Tabel 4.10 Pengujian Tingkat Kesalahan KWh Meter *Master*, dan Tabel 4.11 Pengujian Tingkat Kesalahan KWh Meter *Master*. Nilai ketelitian atau prosentase *error* dari masing-masing KWh Meter ditunjukkan pada tampilan alat uji KWh Meter seperti yang ditunjukkan Gambar 4.10 Tampilan Alat Uji KWh Meter. Pembebanan yang digunakan untuk

pengujian ini yakni pembebanan yang bersifat resistif, yakni pembebanan berupa lampu 100 W. Terdapat tiga KWh meter yang digunakan, yakni : KWh Meter *Master*, KWh Meter Pelanggan A, dan KWh Meter Pelanggan B. Seperti yang tertera pada *nameplate* KWh Meter, ketiga KWh Meter yang digunakan masing – masing termasuk pada Kelas 1. Hal ini yang berarti tingkat kesalahan atau prosentase *error*nya mencapai 1%.



Gambar 4.10 Tampilan Alat Uji KWh Meter

Tabel 4.9 Pengujian Tingkat Kesalahan KWh Meter *Master*

No	Beban	A	V	Tingkat Kesalahan
1	100W	0,43	224,979	0,897%
2	300W	1,36	224,226	0,897%
3	400W	1,79	225,189	0,491%

Tabel 4.10 Pengujian Tingkat Kesalahan KWh Meter Pelanggan A

No	Beban	A	V	Tingkat Kesalahan
1	100W	0,428	223,506	0,044%
2	300W	1,36	225,218	0,089%
3	400W	1,79	225,044	0,089%

Tabel 4.11 Pengujian Tingkat Kesalahan KWh Meter Pelanggan B

No	Beban	A	V	Tingkat Kesalahan
1	100W	0,429	224,191	1,033%
2	300W	1,358	223,269	0,852%
3	400W	1,777	222,755	0,806%

Pengujian Perhitungan KWh Meter ini dilakukan dengan cara menghubungkan KWh meter dengan Watt Meter untuk mengukur beban yang digunakan kemudian dihubungkan kepada beban yang digunakan. Jumlah beban yang digunakan yakni dari rentang 60 Watt hingga 500 Watt. Variabel tetap yang digunakan untuk pengujian KWh Meter ini yakni pada jumlah *impuls* yakni bernilai 5.

Setelah melakukan pengecekan *wiring* antara KWh meter, Watt meter, dan beban kemudian dilakukan pengaturan *timer* yang digunakan untuk mengukur waktu yang dihabiskan oleh tiap beban untuk menghasilkan *impuls* yang berjumlah 5.

Setelah semua persiapan dilakukan maka dilakukan pengujian. Pertama setelah semua *wiring* dinyatakan dalam kondisi benar dan aman kemudian beban dipastikan dalam kondisi *ON*. Kemudian dilakukan pengecekan nilai beban yang ditampilkan oleh watt meter. Setelah itu dilakukan penghitungan mulai dari *impuls* pertama hingga *impuls* kelima. Ketika dilakukan penghitungan *impuls* juga disertai dengan *timer* yang dalam kondisi *ON* untuk melihat berapa banyak waktu yang dihabiskan tiap beban. Setelah *impuls* mencapai pada *impuls* kelima, maka *timer* akan dimatikan. Kemudian dilakukan pencatatan pada Tabel serta perhitungan untuk mendapatkan nilai KW.

Pada KWh Meter yang digunakan untuk pengujian *impuls* ini menghasilkan nilai impuls yang sangat kecil. Maka untuk mengetahui nilai KW didapat dari Persamaan 4.1.

Pada pengujian perhitungan KWh Meter dilakukan sebanyak dua kali, yakni pada KWh meter pelanggan A dan KWh meter pelanggan B. Pada Tabel 4.12 Pengujian Perhitungan KWh Meter Pelanggan A telah dijelaskan mengenai pengujian perhitungan KWh Meter pada pelanggan A. Pengujian KWh meter pada pelanggan B dijelaskan pada Tabel 4.13 Pengujian Perhitungan KWh Meter Pelanggan B. pembebanan yang digunakan yakni berupa pembebanan resistif berupa lampu dengan rentang daya yakni 60 W hingga 100 W. kemudian nilai daya pada lampu tersebut ditunjukkan oleh Watt Meter. *Impuls* yang dihasilkan dari daya yang telah dihubungkan dengan kWh meter tersebut

dihitung sebanyak 5 *impuls* kemudian dicatat sesuai lamanya waktu sesuai dengan 5 *impuls* yang telah terhitung.

Tabel 4.12 Pengujian Perhitungan KWh Meter Pelanggan A

No	Beban (W)	Beban pada Watt Meter (W)	Jumlah <i>Impuls</i>	Waktu (Detik)	Nilai kW
1	60	50	5	111	0,05
2	100	90	5	65	0,08
3	100	100	5	55	0,10
4	125	115	5	50	0,11
5	150	140	5	40	0,14
6	175	165	5	34	0,16
7	200	200	5	28	0,20
8	300	300	5	18	0,31
9	400	395	5	14	0,40
10	500	495	5	11	0,51

Tabel 4.13 Pengujian Perhitungan KWh Meter Pelanggan B

No	Beban (W)	Beban pada Watt Meter (W)	Jumlah <i>Impuls</i>	Waktu (Detik)	Nilai kW
1	60	55	5	113	0,05
2	100	90	5	65	0,08
3	100	100	5	56	0,10
4	125	115	5	50	0,11
5	150	135	5	41	0,14
6	175	165	5	35	0,16
7	200	200	5	28	0,20
8	300	300	5	18	0,31
9	400	400	5	14	0,40
10	500	500	5	11	0,51

4.4 Pengujian *Prototype Simulator* (Sistem Keseluruhan)

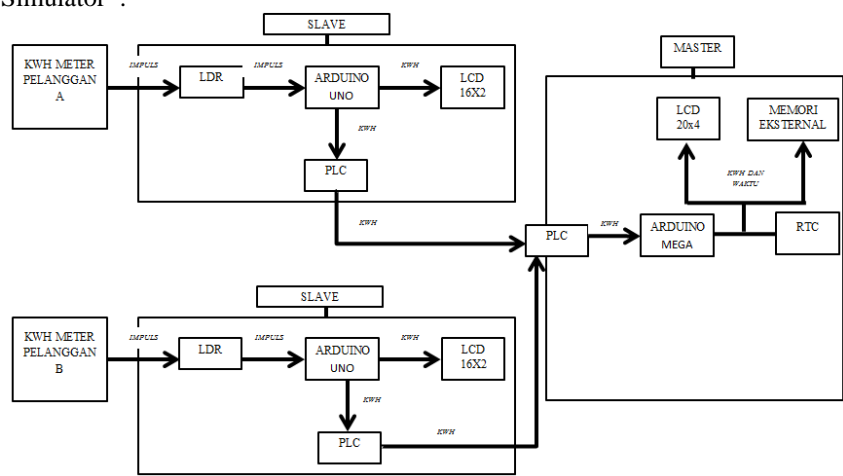
Pengujian *Prototype* Seimulator merupakan pengujian secara serentak terhadap KWh Meter, *Slave* dan *Master*. Pengujian ini dapat dilihat pada Gambar 4.11 Pengujian *Prototype Simulator*. Pada bagian *Slave*, pengujian terhadap pengukuran kWh meter dengan menggunakan

perhitungan *impuls*. *Impuls* yang dihasilkan oleh KWh meter dideteksi menggunakan sensor LDR. *Impuls* tersebut lalu dikonversikan terhadap satuan kWh. Proses pengkonversian dari satuan *impuls* menjadi satuan kWh melalui pemrograman terhadap arduino uno yang terdapat pada *Slave*. Hasil pengukuran tersebut lalu akan ditampilkan pada LCD 16 x 2. Baik untuk pengukuran KW pada pelanggan A maupun pada pelanggan B. Selain data pengukuran kWh tersebut ditampilkan pada LCD juga diproses dengan arduino uno untuk dikirim terhadap *Master* yang terdapat pada *prototype* panel PHB melalui *media Power Line Carrier*. *Power Line Carrier* berfungsi sebagai media pengiriman serta penerimaan data antara *Slave* dan *Master*.

Setelah *Master* menerima data kWh yang telah dikirim oleh *Slave*, baik data kWh pelanggan A maupun data kWh Pelanggan B. Data kWh yang dikirimkan secara berkala oleh *Slave* tersebut kemudian disimpan oleh EEPROM yang kemudian untuk ditampilkan pada LCD 20 x 4 beserta dengan waktu penerimaannya. LCD untuk melihat data kWh Pelanggan A ataupun data kWh Pelanggan B yang tercatat beserta waktu pencatatannya. Dengan menekan *Push Button* juga dapat mengalihkan tampilan LCD menjadi menampilkan selisih kWh antara kWh yang digunakan oleh pelanggan pada bulan kemarin dan juga pemakaian kWh bulan ini. Pencatatan waktu pengiriman data kWh yang kemudian ditampilkan oleh LCD juga akan disimpan pada memori kartu (SD Card) merupakan fungsi digunakannya RTC pada rangkaian *Master* ini. Data kWh yang telah dikirim oleh *Slave* secara berkala pada waktu-waktu tertentu tersebut kemudian disimpan pada memori berbentuk kartu yang telah disediakan pada *Master* guna mempermudah pencatatan penggunaan kWh yang ada pada pelanggan. Program pengujian untuk *prototype* simulator dapat dilihat pada Lampiran A.1 Pemrograman *Slave* untuk pada sisi modul *Slave* yakni pada sisi pelanggan dan Lampiran A.4 Pemrograman *Master* untuk pada sisi modul *Master* yakni tiang PHB PLN.

Pada Tabel 4.14 Pengujian *Prototype* Simulator, jumlah beban yang digunakan yakni dari rentang 60 Watt hingga 500 Watt. Beban yang digunakan berupa lampu yang bersifat pembebanan resistif. Variabel tetap yang digunakan untuk pengujian *Prototype* Simulator ini yakni pada jumlah *impuls* yakni bernilai 5. Dengan nilai ketelitian yang berbeda antara KWh Meter pelanggan A dan KWh Meter Pelanggan B mempengaruhi pengukuran kWh pada masing-masing KWh Meter yang telah digunakan oleh pelanggan. Sehingga pada nilai beban tertentu

terdapat perbedaan nilai pengukuran yang sedikit mencolok. Setelah data pengukuran kWh dikirim oleh *Slave* dan diterima oleh *Master*, maka nilai keseluruhan kWh pada pelanggan yang menggunakan fasa yang sama dapat diketahui. Data penggunaan kWh masing-masing pelanggan dan nilai keseluruhan kWh yang digunakan oleh pelanggan dapat diketahui melalui tampilan LCD 20 X 4 yang terdapat pada *Master*. data penggunaan kWh masing-masing pelanggan tersebut kemudian disimpan pada memori kartu yang telah disediakan dan dilengkapi dengan catatan penerimaan data kWh. Perbedaan waktu pengiriman antara pelanggan A dengan pelanggan B terdapat selisih waktu 2 detik. Hal ini ditunjukkan pada Tabel 4.15 Pengujian Waktu pada *Prototype Simulator* .



Gambar 4.11 Pengujian *Prototype Simulator*

Tabel 4.14 Pengujian *Prototype Simulator*

No	Beban (W)	Jarak PLC (m)	Slave		Master		
			Pelanggan A (KWh)	Pelanggan B (KWh)	Waktu	Memori Pelanggan A	Memori Pelanggan B
1	60	5	180	180	10.00	180	180

2	100	5	288	288	10.00	288	288
3	100	5	360	360	10.00	360	360
4	125	5	396	396	10.00	396	396
5	150	5	504	504	10.00	504	504
6	175	5	576	576	10.00	576	576
7	200	5	720	720	10.00	720	720
8	300	5	1116	1116	10.00	1116	1116
9	400	5	1440	1440	10.00	1440	1440
10	500	5	1836	1836	10.00	1836	1836

Tabel 4.15 Pengujian Waktu pada *Prototype Simulator*

No	Beban (W)	Jarak PLC (m)	Slave		Master	
			Pelanggan A (kWh)	Pelanggan B (kWh)	Waktu Pelanggan A	Waktu Pelanggan B
1	60	5	180	180	10.00.00	10.00.02
2	100	5	288	288	10.00.00	10.00.02
3	100	5	360	360	10.00.00	10.00.02
4	125	5	396	396	10.00.00	10.00.02
5	150	5	504	504	10.00.00	10.00.02
6	175	5	576	576	10.00.00	10.00.02
7	200	5	720	720	10.00.00	10.00.02
8	300	5	1116	1116	10.00.00	10.00.02
9	400	5	1440	1440	10.00.00	10.00.02
10	500	5	1836	1836	10.00.00	10.00.02

--- Halaman ini sengaja dikosongkan ---

BAB V

PENUTUP

Setelah melakukan perencanaan dan pembuatan alat serta pengujian dan analisis, maka dapat ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan.

5.1 Kesimpulan

Dari Tugas Akhir yang telah dikerjakan, dapat disimpulkan bahwa:

1. Pengukuran KWh meter dengan media *Power Line Carrier* dapat dilakukan dengan metode komunikasi serial dapat dilakukan dalam jarak teruji hingga 35 meter.
2. Komunikasi pengiriman data menggunakan PLC lebih sedikit mengalami *error* ketika rentang waktu tiap pengiriman pendek hendaknya diberi sedikit tenggang waktu atau *delay* yakni 2 detik.
3. Pengukuran KWh Meter digital dapat dilakukan dengan mendeteksi *impuls* yang telah dihasilkan oleh KWh Meter Digital melalui sensor *Light Dependent Resistor*.
4. Dengan menghimpun pengukuran nilai kWh pada trafo tiang, dapat diketahui selisih pengukuran kWh meter antara kWh yang telah diukur pada kWh meter pelanggan dengan pengukuran kWh meter pada trafo tiang.
5. Dalam pengiriman menggunakan media *power line carrier* hendaknya dilakukan secara bergiliran sehingga menghindari *error* saat penerimaan data yang dikirim serta memudahkan pencatatanya
6. Tingkat kesalahan pengukuran KWh Meter sebesar kurang lebih 1%. Hal ini dikarenakan KWh meter yang digunakan merupakan kelas 1.

5.2 Saran

Diberikan beberapa saran yang sekiranya dapat dikembangkan pada masa yang akan datang demi kesempurnaan dari proyek tugas akhir ini. Adapun beberapa saran tersebut adalah sebagai berikut :

1. *Power Line Carrier* (PLC) yang digunakan peengukuran KWh meter pelanggan sebaiknya di kemas menjadi satu kesatuan dengan KWh meter milik pelanggan untuk menghindarai kerusakan dan juga terjadinya kecurangan di pelanggan.
2. Untuk menghasilkan pengukuran kWh meter pelanggan

dengan ketelitian lebih tinggi sebaiknya menggunakan KWh Meter dengan kelas ketelitian lebih dari Kelas 1.

DAFTAR PUSTAKA

- [1], SPLN 69-2 :1987 : **Standardisasi Peralatan Uji**, PT. PLN (Persero) Pusat Pendidikan dan Pelatihan, Jakarta, 1987.
- [2] Aji Pudya Khaira dan Dany Saifuddin, “Purwarupa Sistem Monitoring Pemakaian Daya pada Tiap – Tiap Rumah dengan *Power Line Carrier* Berbasis Mikrokontroller”, **Tugas Akhir**, Jurusan D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya, 2015.
- [3] Fahrul, Marzuki, “Aplikasi *Power Line Carrier* (PLC) Untuk Komunikasi Pada Daerah”, **Jurnal Teknik**, Fakultas Teknik, Universitas Jakarta, Jakarta, 2008.
- [4] Diana Gita Andriana Putri dan Rendy Nur Hidayatullah, “Monitoring Tegangan dan Arus pada Battery Housing Menggunakan Mikrokontroller dan WIFI”, **Tugas Akhir**, Jurusan D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.
- [5] Suwanto, Eko Andy, “Rancang Bangun Meter KWH Prabayar Pelanggan Fasa Tunggal Melalui PLC (*Power Line Carrier*) Berbasis Mikrokontroller”, **Tugas Akhir**, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya, 2007.
- [6] Nafita Hana dan Rifki Nurafian, “Smart Electric Meter”, **Tugas Akhir**, Jurusan D3 Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.

--- Halaman ini sengaja dikosongkan ---

LAMPIRAN A

A.1 Pemrograman *Slave*

```
#include <EEPROM.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(10, 9, 8, 7, 6, 5);
SoftwareSerial Uno (4,3);
int tombol=11;
int
v0,v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v
19,v20;
int
v21,v22,v23,v24,v25,v26,v27,v28,v29,v30,v31,v32,v33,v34,v35,v36,v3
7,v38,v39,v40;
int
v41,v42,v43,v44,v45,v46,v47,v48,v49,v50,v51,v52,v53,v54,v55,v56,v5
7,v58,v59,v60;
int
v61,v62,v63,v64,v65,v66,v67,v68,v69,v70,v71,v72,v73,v74,v75,v76,v7
7,v78,v79,v80;
int
v81,v82,v83,v84,v85,v86,v87,v88,v89,v90,v91,v92,v93,v94,v95,v96,v9
7,v98,v99,v100;
int v255,v254,v253,v252,v251,v250,v249,v248,v247,pulrib=0, rib=0,
rat=0, pul=0, sat=0; ;
int ldr=A0, i=0, ldr_n=21, kedip=21,kWh;
float kWhbgi,kWh2, kWhbg;
void setup()
{
  lcd.begin(16, 2);
  lcd.print("Plgn ke 1");
  v0 = EEPROM.read(0);v1 = EEPROM.read(1);v2 =
  EEPROM.read(2);v3 = EEPROM.read(3);v4 = EEPROM.read(4);
  v5 = EEPROM.read(5);v6 = EEPROM.read(6);v7 =
  EEPROM.read(7);v8 = EEPROM.read(8);v9 = EEPROM.read(9);
  v10 = EEPROM.read(10);v11 = EEPROM.read(11);v12 =
  EEPROM.read(12);v13 = EEPROM.read(13);v14 = EEPROM.read(14);
```

```
v15 = EEPROM.read(15);v16 = EEPROM.read(16);v17 =  
EEPROM.read(17);v18 = EEPROM.read(18);v19 = EEPROM.read(19);  
v20 = EEPROM.read(20);v21 = EEPROM.read(21);v22 =  
EEPROM.read(22);v23 = EEPROM.read(23);v24 = EEPROM.read(24);  
v25 = EEPROM.read(25);v26 = EEPROM.read(26);v27 =  
EEPROM.read(27);v28 = EEPROM.read(28);v29 = EEPROM.read(29);  
v30 = EEPROM.read(30);v31 = EEPROM.read(31);v32 =  
EEPROM.read(32);v33 = EEPROM.read(33);v34 = EEPROM.read(34);  
v35 = EEPROM.read(35);v36 = EEPROM.read(36);v37 =  
EEPROM.read(37);v38 = EEPROM.read(38);v39 = EEPROM.read(39);  
v40 = EEPROM.read(40);v41 = EEPROM.read(41);v42 =  
EEPROM.read(42);v43 = EEPROM.read(43);v44 = EEPROM.read(44);  
v45 = EEPROM.read(45);v46 = EEPROM.read(46);v47 =  
EEPROM.read(47);v48 = EEPROM.read(48);v49 = EEPROM.read(49);  
v50 = EEPROM.read(50);v51 = EEPROM.read(51);v52 =  
EEPROM.read(52);v53 = EEPROM.read(53);v54 = EEPROM.read(54);  
v55 = EEPROM.read(55);v56 = EEPROM.read(56);v57 =  
EEPROM.read(57);v58 = EEPROM.read(58);v59 = EEPROM.read(59);  
v60 = EEPROM.read(60);v61 = EEPROM.read(61);v62 =  
EEPROM.read(62);v63 = EEPROM.read(63);v64 = EEPROM.read(64);  
v65 = EEPROM.read(65);v66 = EEPROM.read(66);v67 =  
EEPROM.read(67);v68 = EEPROM.read(68);v69 = EEPROM.read(69);  
v70 = EEPROM.read(70);v71 = EEPROM.read(71);v72 =  
EEPROM.read(72);v73 = EEPROM.read(73);v74 = EEPROM.read(74);  
v75 = EEPROM.read(75);v76 = EEPROM.read(76);v77 =  
EEPROM.read(77);v78 = EEPROM.read(78);v79 = EEPROM.read(79);  
v80 = EEPROM.read(80);v81 = EEPROM.read(81);v82 =  
EEPROM.read(82);v83 = EEPROM.read(83);v84 = EEPROM.read(84);  
v85 = EEPROM.read(85);v86 = EEPROM.read(86);v87 =  
EEPROM.read(87);v88 = EEPROM.read(88);v89 = EEPROM.read(89);  
v90 = EEPROM.read(90);v91 = EEPROM.read(91);v92 =  
EEPROM.read(92);v93 = EEPROM.read(93);v94 = EEPROM.read(94);  
v95 = EEPROM.read(95);v96 = EEPROM.read(96);v97 =  
EEPROM.read(97);v98 = EEPROM.read(98);v99 = EEPROM.read(99);  
v100 = EEPROM.read(100);v255 = EEPROM.read(255);v254 =  
EEPROM.read(254);v253 = EEPROM.read(253);v252 =  
EEPROM.read(252);
```

```

v251 = EEPROM.read(251);v250 = EEPROM.read(250);v249 =
EEPROM.read(249);v248 = EEPROM.read(248);v247 =
EEPROM.read(247);
if (v0!=1){kedip=0;}else if (v0==1 and v1!=1){kedip=32;kWh2=1;}
else if (v1==1 and v2!=1){kedip=64;kWh2=2;}else if (v2==1 and
v3!=1){kedip=96;kWh2=3;}
else if (v3==1 and v4!=1){kedip=128;kWh2=4;}else if (v4==1 and
v5!=1){kedip=160;kWh2=5;}
else if (v5==1 and v6!=1){kedip=192;kWh2=6;}else if (v6==1 and
v7!=1){kedip=224;kWh2=7;}
else if (v7==1 and v8!=1){kedip=256;kWh2=8;}else if (v8==1 and
v9!=1){kedip=288;kWh2=9;}
else if (v9==1 and v10!=1){kedip=320;kWh2=10;}else if (v10==1 and
v11!=1){kedip=352;kWh2=11;}
else if (v11==1 and v12!=1){kedip=384;kWh2=12;}else if (v12==1 and
v13!=1){kedip=416;kWh2=13;}
else if (v13==1 and v14!=1){kedip=448;kWh2=14;}else if (v14==1 and
v15!=1){kedip=480;kWh2=15;}
else if (v15==1 and v16!=1){kedip=512;kWh2=16;}else if (v16==1 and
v17!=1){kedip=544;kWh2=17;}
else if (v17==1 and v18!=1){kedip=576;kWh2=18;}else if (v18==1 and
v19!=1){kedip=608;kWh2=19;}
else if (v19==1 and v20!=1){kedip=640;kWh2=20;}else if (v20==1 and
v21!=1){kedip=672;kWh2=21;}
else if (v21==1 and v22!=1){kedip=704;kWh2=22;}else if (v22==1 and
v23!=1){kedip=736;kWh2=23;}
else if (v23==1 and v24!=1){kedip=768;kWh2=24;}else if (v24==1 and
v25!=1){kedip=800;kWh2=25;}
else if (v25==1 and v26!=1){kedip=832;kWh2=26;}else if (v26==1 and
v27!=1){kedip=864;kWh2=27;}
else if (v27==1 and v28!=1){kedip=896;kWh2=28;}else if (v28==1 and
v29!=1){kedip=928;kWh2=29;}
else if (v29==1 and v30!=1){kedip=960;kWh2=30;}else if (v30==1 and
v31!=1){kedip=992;kWh2=31;}
else if (v31==1 and v32!=1){kedip=1024;kWh2=32;}else if (v32==1
and v33!=1){kedip=1056;kWh2=33;}
else if (v33==1 and v34!=1){kedip=1088;kWh2=34;}else if (v34==1
and v35!=1){kedip=1120;kWh2=35;}

```

```

else if (v35==1 and v36!=1){kedi p=1152;kWh2=36;}else if (v36==1
and v37!=1){kedi p=1184;kWh2=37;}
else if (v37==1 and v38!=1){kedi p=1216;kWh2=38;}else if (v38==1
and v39!=1){kedi p=1248;kWh2=39;}
else if (v39==1 and v40!=1){kedi p=1280;kWh2=40;}else if (v40==1
and v41!=1){kedi p=1312;kWh2=41;}
else if (v41==1 and v42!=1){kedi p=1344;kWh2=42;}else if (v42==1
and v43!=1){kedi p=1376;kWh2=43;}
else if (v43==1 and v44!=1){kedi p=1408;kWh2=44;}else if (v44==1
and v45!=1){kedi p=1440;kWh2=45;}
else if (v45==1 and v46!=1){kedi p=1472;kWh2=46;}else if (v46==1
and v47!=1){kedi p=1504;kWh2=47;}
else if (v47==1 and v48!=1){kedi p=1536;kWh2=48;}else if (v48==1
and v49!=1){kedi p=1568;kWh2=49;}
else if (v49==1 and v50!=1){kedi p=1600;kWh2=50;}else if (v50==1
and v51!=1){kedi p=1632;kWh2=51;}
else if (v51==1 and v52!=1){kedi p=1664;kWh2=52;}else if (v52==1
and v53!=1){kedi p=1696;kWh2=53;}
else if (v53==1 and v54!=1){kedi p=1728;kWh2=54;}else if (v54==1
and v55!=1){kedi p=1760;kWh2=55;}
else if (v55==1 and v56!=1){kedi p=1792;kWh2=56;}else if (v56==1
and v57!=1){kedi p=1824;kWh2=57;}
else if (v57==1 and v58!=1){kedi p=1856;kWh2=58;}else if (v58==1
and v59!=1){kedi p=1888;kWh2=59;}
else if (v59==1 and v60!=1){kedi p=1920;kWh2=60;}else if (v60==1
and v61!=1){kedi p=1952;kWh2=61;}
else if (v61==1 and v62!=1){kedi p=1984;kWh2=62;}else if (v62==1
and v63!=1){kedi p=2016;kWh2=63;}
else if (v63==1 and v64!=1){kedi p=2048;kWh2=64;}else if (v64==1
and v65!=1){kedi p=2080;kWh2=65;}
else if (v65==1 and v66!=1){kedi p=2112;kWh2=66;}else if (v66==1
and v67!=1){kedi p=2144;kWh2=67;}
else if (v67==1 and v68!=1){kedi p=2176;kWh2=68;}else if (v68==1
and v69!=1){kedi p=2208;kWh2=69;}
else if (v69==1 and v70!=1){kedi p=2240;kWh2=70;}else if (v70==1
and v71!=1){kedi p=2272;kWh2=71;}
else if (v71==1 and v72!=1){kedi p=2304;kWh2=72;}else if (v72==1
and v73!=1){kedi p=2336;kWh2=73;}

```

```

else if (v73==1 and v74!=1){kedi p=2368;kWh2=74;}else if (v74==1
and v75!=1){kedi p=2400;kWh2=75;}
else if (v75==1 and v76!=1){kedi p=2432;kWh2=76;}else if (v76==1
and v77!=1){kedi p=2464;kWh2=77;}
else if (v77==1 and v78!=1){kedi p=2496;kWh2=78;}else if (v78==1
and v79!=1){kedi p=2528;kWh2=79;}
else if (v79==1 and v80!=1){kedi p=2560;kWh2=80;}else if (v80==1
and v81!=1){kedi p=2592;kWh2=81;}
else if (v81==1 and v82!=1){kedi p=2624;kWh2=82;}else if (v82==1
and v83!=1){kedi p=2656;kWh2=83;}
else if (v83==1 and v84!=1){kedi p=2688;kWh2=84;}else if (v84==1
and v85!=1){kedi p=2720;kWh2=85;}
else if (v85==1 and v86!=1){kedi p=2752;kWh2=86;}else if (v86==1
and v87!=1){kedi p=2784;kWh2=87;}
else if (v87==1 and v88!=1){kedi p=2816;kWh2=88;}else if (v88==1
and v89!=1){kedi p=2848;kWh2=89;}
else if (v89==1 and v90!=1){kedi p=2880;kWh2=90;}else if (v90==1
and v91!=1){kedi p=2912;kWh2=91;}
else if (v91==1 and v92!=1){kedi p=2944;kWh2=92;}else if (v92==1
and v93!=1){kedi p=2976;kWh2=93;}
else if (v93==1 and v94!=1){kedi p=3008;kWh2=94;}else if (v94==1
and v95!=1){kedi p=3040;kWh2=95;}
else if (v95==1 and v96!=1){kedi p=3072;kWh2=96;}else if (v96==1
and v97!=1){kedi p=3104;kWh2=97;}
else if (v97==1 and v98!=1){kedi p=3136;kWh2=98;}else if (v98==1
and v99!=1){kedi p=3168;kWh2=99;}
Serial.begin (9600);Uno.begin(9600);
}
void loop()
{
lcd.setCursor(0, 1);
lcd.print(kWh);
lcd.print(".");
if (kWh2<10){lcd.print("0");}
lcd.print(kWh2);
lcd.print("KWh");
lcd.print(kedi p);
//ldr_n=analogRead(ldr);
if (ldr_n >= 20)

```

```

{kedip=kedip+1;
if (kedip%32==0){kWh2=kWh2+1;}
kWh = (v255*10000) + (v254*1000) + (v253*100) +
(v252*10)+v251;
//ldr_n=analogRead(ldr);
if (ldr_n >= 20)
{kedip=kedip+1;
if (v0!=1 and kedip%32==0){EEPROM.write(0,1);kWh2=kWh2+1;}
if (v0==1 and v1!=1 and
kedip%32==0){EEPROM.write(1,1);kWh2=kWh2+1;}
if (v1==1 and v2!=1 and
kedip%32==0){EEPROM.write(2,1);kWh2=kWh2+1;}
if (v2==1 and v3!=1 and
kedip%32==0){EEPROM.write(3,1);kWh2=kWh2+1;}
if (v3==1 and v4!=1 and
kedip%32==0){EEPROM.write(4,1);kWh2=kWh2+1;}
if (v4==1 and v5!=1 and
kedip%32==0){EEPROM.write(5,1);kWh2=kWh2+1;}
if (v5==1 and v6!=1 and
kedip%32==0){EEPROM.write(6,1);kWh2=kWh2+1;}
if (v6==1 and v7!=1 and
kedip%32==0){EEPROM.write(7,1);kWh2=kWh2+1;}
if (v7==1 and v8!=1 and
kedip%32==0){EEPROM.write(8,1);kWh2=kWh2+1;}
if (v8==1 and v9!=1 and
kedip%32==0){EEPROM.write(9,1);kWh2=kWh2+1;}
if (v9==1 and v10!=1 and
kedip%32==0){EEPROM.write(10,1);kWh2=kWh2+1;}
if (v10==1 and v11!=1 and
kedip%32==0){EEPROM.write(11,1);kWh2=kWh2+1;}
if (v11==1 and v12!=1 and
kedip%32==0){EEPROM.write(12,1);kWh2=kWh2+1;}
if (v12==1 and v13!=1 and
kedip%32==0){EEPROM.write(13,1);kWh2=kWh2+1;}
if (v13==1 and v14!=1 and
kedip%32==0){EEPROM.write(14,1);kWh2=kWh2+1;}
if (v14==1 and v15!=1 and
kedip%32==0){EEPROM.write(15,1);kWh2=kWh2+1;}

```

```

    if (v15==1 and v16!=1 and
kedip%32==0){EEPROM.write(16,1);kWh2=kWh2+1;}
    if (v16==1 and v17!=1 and
kedip%32==0){EEPROM.write(17,1);kWh2=kWh2+1;}
    if (v17==1 and v18!=1 and
kedip%32==0){EEPROM.write(18,1);kWh2=kWh2+1;}
    if (v18==1 and v19!=1 and
kedip%32==0){EEPROM.write(19,1);kWh2=kWh2+1;}
    if (v19==1 and v20!=1 and
kedip%32==0){EEPROM.write(20,1);kWh2=kWh2+1;}
    if (v20==1 and v21!=1 and
kedip%32==0){EEPROM.write(21,1);kWh2=kWh2+1;}
    if (v21==1 and v22!=1 and
kedip%32==0){EEPROM.write(22,1);kWh2=kWh2+1;}
    if (v22==1 and v23!=1 and
kedip%32==0){EEPROM.write(23,1);kWh2=kWh2+1;}
    if (v23==1 and v24!=1 and
kedip%32==0){EEPROM.write(24,1);kWh2=kWh2+1;}
    if (v24==1 and v25!=1 and
kedip%32==0){EEPROM.write(25,1);kWh2=kWh2+1;}
    if (v25==1 and v26!=1 and
kedip%32==0){EEPROM.write(26,1);kWh2=kWh2+1;}
    if (v26==1 and v27!=1 and
kedip%32==0){EEPROM.write(27,1);kWh2=kWh2+1;}
    if (v27==1 and v28!=1 and
kedip%32==0){EEPROM.write(28,1);kWh2=kWh2+1;}
    if (v28==1 and v29!=1 and
kedip%32==0){EEPROM.write(29,1);kWh2=kWh2+1;}
    if (v29==1 and v30!=1 and
kedip%32==0){EEPROM.write(30,1);kWh2=kWh2+1;}
    if (v30==1 and v31!=1 and
kedip%32==0){EEPROM.write(31,1);kWh2=kWh2+1;}
    if (v31==1 and v32!=1 and
kedip%32==0){EEPROM.write(32,1);kWh2=kWh2+1;}
    if (v32==1 and v33!=1 and
kedip%32==0){EEPROM.write(33,1);kWh2=kWh2+1;}
    if (v33==1 and v34!=1 and
kedip%32==0){EEPROM.write(34,1);kWh2=kWh2+1;}

```

```

    if (v34==1 and v35!=1 and
kedip%32==0){EEPROM.write(35,1);kWh2=kWh2+1;}
    if (v35==1 and v36!=1 and
kedip%32==0){EEPROM.write(36,1);kWh2=kWh2+1;}
    if (v36==1 and v37!=1 and
kedip%32==0){EEPROM.write(37,1);kWh2=kWh2+1;}
    if (v37==1 and v38!=1 and
kedip%32==0){EEPROM.write(38,1);kWh2=kWh2+1;}
    if (v38==1 and v39!=1 and
kedip%32==0){EEPROM.write(39,1);kWh2=kWh2+1;}
    if (v39==1 and v40!=1 and
kedip%32==0){EEPROM.write(40,1);kWh2=kWh2+1;}
    if (v40==1 and v41!=1 and
kedip%32==0){EEPROM.write(41,1);kWh2=kWh2+1;}
    if (v41==1 and v42!=1 and
kedip%32==0){EEPROM.write(42,1);kWh2=kWh2+1;}
    if (v42==1 and v43!=1 and
kedip%32==0){EEPROM.write(43,1);kWh2=kWh2+1;}
    if (v43==1 and v44!=1 and
kedip%32==0){EEPROM.write(44,1);kWh2=kWh2+1;}
    if (v44==1 and v45!=1 and
kedip%32==0){EEPROM.write(45,1);kWh2=kWh2+1;}
    if (v45==1 and v46!=1 and
kedip%32==0){EEPROM.write(46,1);kWh2=kWh2+1;}
    if (v46==1 and v47!=1 and
kedip%32==0){EEPROM.write(47,1);kWh2=kWh2+1;}
    if (v47==1 and v48!=1 and
kedip%32==0){EEPROM.write(48,1);kWh2=kWh2+1;}
    if (v48==1 and v49!=1 and
kedip%32==0){EEPROM.write(49,1);kWh2=kWh2+1;}
    if (v49==1 and v50!=1 and
kedip%32==0){EEPROM.write(50,1);kWh2=kWh2+1;}
    if (v50==1 and v51!=1 and
kedip%32==0){EEPROM.write(51,1);kWh2=kWh2+1;}
    if (v51==1 and v52!=1 and
kedip%32==0){EEPROM.write(52,1);kWh2=kWh2+1;}
    if (v52==1 and v53!=1 and
kedip%32==0){EEPROM.write(53,1);kWh2=kWh2+1;}

```



```

    if (v53==1 and v54!=1 and
kedip%32==0){EEPROM.write(54,1);kWh2=kWh2+1;}
    if (v54==1 and v55!=1 and
kedip%32==0){EEPROM.write(55,1);kWh2=kWh2+1;}
    if (v55==1 and v56!=1 and
kedip%32==0){EEPROM.write(56,1);kWh2=kWh2+1;}
    if (v56==1 and v57!=1 and
kedip%32==0){EEPROM.write(57,1);kWh2=kWh2+1;}
    if (v57==1 and v58!=1 and
kedip%32==0){EEPROM.write(58,1);kWh2=kWh2+1;}
    if (v58==1 and v59!=1 and
kedip%32==0){EEPROM.write(59,1);kWh2=kWh2+1;}
    if (v59==1 and v60!=1 and
kedip%32==0){EEPROM.write(60,1);kWh2=kWh2+1;}
    if (v60==1 and v61!=1 and
kedip%32==0){EEPROM.write(61,1);kWh2=kWh2+1;}
    if (v61==1 and v62!=1 and
kedip%32==0){EEPROM.write(62,1);kWh2=kWh2+1;}
    if (v62==1 and v63!=1 and
kedip%32==0){EEPROM.write(63,1);kWh2=kWh2+1;}
    if (v63==1 and v64!=1 and
kedip%32==0){EEPROM.write(64,1);kWh2=kWh2+1;}
    if (v64==1 and v65!=1 and
kedip%32==0){EEPROM.write(65,1);kWh2=kWh2+1;}
    if (v65==1 and v66!=1 and
kedip%32==0){EEPROM.write(66,1);kWh2=kWh2+1;}
    if (v66==1 and v67!=1 and
kedip%32==0){EEPROM.write(67,1);kWh2=kWh2+1;}
    if (v67==1 and v68!=1 and
kedip%32==0){EEPROM.write(68,1);kWh2=kWh2+1;}
    if (v68==1 and v69!=1 and
kedip%32==0){EEPROM.write(69,1);kWh2=kWh2+1;}
    if (v69==1 and v70!=1 and
kedip%32==0){EEPROM.write(70,1);kWh2=kWh2+1;}
    if (v70==1 and v71!=1 and
kedip%32==0){EEPROM.write(71,1);kWh2=kWh2+1;}
    if (v71==1 and v72!=1 and
kedip%32==0){EEPROM.write(72,1);kWh2=kWh2+1;}

```

```

    if (v72==1 and v73!=1 and
kedip%32==0){EEPROM.write(73,1);kWh2=kWh2+1;}
    if (v73==1 and v74!=1 and
kedip%32==0){EEPROM.write(74,1);kWh2=kWh2+1;}
    if (v74==1 and v75!=1 and
kedip%32==0){EEPROM.write(75,1);kWh2=kWh2+1;}
    if (v75==1 and v76!=1 and
kedip%32==0){EEPROM.write(76,1);kWh2=kWh2+1;}
    if (v76==1 and v77!=1 and
kedip%32==0){EEPROM.write(77,1);kWh2=kWh2+1;}
    if (v77==1 and v78!=1 and
kedip%32==0){EEPROM.write(78,1);kWh2=kWh2+1;}
    if (v78==1 and v79!=1 and
kedip%32==0){EEPROM.write(79,1);kWh2=kWh2+1;}
    if (v79==1 and v80!=1 and
kedip%32==0){EEPROM.write(80,1);kWh2=kWh2+1;}
    if (v80==1 and v81!=1 and
kedip%32==0){EEPROM.write(81,1);kWh2=kWh2+1;}
    if (v81==1 and v82!=1 and
kedip%32==0){EEPROM.write(82,1);kWh2=kWh2+1;}
    if (v82==1 and v83!=1 and
kedip%32==0){EEPROM.write(83,1);kWh2=kWh2+1;}
    if (v83==1 and v84!=1 and
kedip%32==0){EEPROM.write(84,1);kWh2=kWh2+1;}
    if (v84==1 and v85!=1 and
kedip%32==0){EEPROM.write(85,1);kWh2=kWh2+1;}
    if (v85==1 and v86!=1 and
kedip%32==0){EEPROM.write(86,1);kWh2=kWh2+1;}
    if (v86==1 and v87!=1 and
kedip%32==0){EEPROM.write(87,1);kWh2=kWh2+1;}
    if (v87==1 and v88!=1 and
kedip%32==0){EEPROM.write(88,1);kWh2=kWh2+1;}
    if (v88==1 and v89!=1 and
kedip%32==0){EEPROM.write(89,1);kWh2=kWh2+1;}
    if (v89==1 and v90!=1 and
kedip%32==0){EEPROM.write(90,1);kWh2=kWh2+1;}
    if (v90==1 and v91!=1 and
kedip%32==0){EEPROM.write(91,1);kWh2=kWh2+1;}

```

```

    if (v91==1 and v92!=1 and
kedip%32==0){EEPROM.write(92,1);kWh2=kWh2+1;}
    if (v92==1 and v93!=1 and
kedip%32==0){EEPROM.write(93,1);kWh2=kWh2+1;}
    if (v93==1 and v94!=1 and
kedip%32==0){EEPROM.write(94,1);kWh2=kWh2+1;}
    if (v94==1 and v95!=1 and
kedip%32==0){EEPROM.write(95,1);kWh2=kWh2+1;}
    if (v95==1 and v96!=1 and
kedip%32==0){EEPROM.write(96,1);kWh2=kWh2+1;}
    if (v96==1 and v97!=1 and
kedip%32==0){EEPROM.write(97,1);kWh2=kWh2+1;}
    if (v97==1 and v98!=1 and
kedip%32==0){EEPROM.write(98,1);kWh2=kWh2+1;}
    if (v98==1 and v99!=1 and
kedip%32==0){EEPROM.write(99,1);kWh2=kWh2+1;}
    if (v99==1 and v100!=1 and kedip%32==0){kWh=kWh + 1 ;}
    if (kWh2==100){kWh2=0;kWh=kWh+1;
    pulrib = (kWh%100000)/10000;if(pulrib!=0 and
pulrib!=v255){v255=pulrib;delay(10);}
    rib = (kWh%10000)/1000; if(rib!=0 and
rib!=v254){v254=rib;delay(10);}
    rat = (kWh%1000)/100;if((rib!=0 or rat!=0) and rat!=v253
){v253=rat; delay(10);}
    pul = (kWh%100)/10;if((rib!=0 or rat!=0 or pul!=0) and
pul!=v252){v252=pul; delay(10);}
    sat = (kWh%10);if((rib!=0 or rat!=0 or pul!=0 or sat!=0) and
sat!=v252){v251=sat; delay(10);}
    kWh2=0;
    kedip=0;
    Uno.write("a");

else
{kedip = kedip;}
}
}

```

A.2 Pemrograman LCD 2X16

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(10, 9, 8, 7, 6, 5);
```

```
void setup() {
```

```
    // set up the LCD's number of columns and rows:
```

```
    lcd.begin(16, 2);
```

```
    // Print a message to the LCD.
```

```
    lcd.print("hello, world!");
```

```
}
```

```
void loop() {
```

```
    // set the cursor to column 0, line 1
```

```
    // (note: line 1 is the second row, since counting begins with 0):
```

```
    lcd.setCursor(0, 1);
```

```
    // print the number of seconds since reset:
```

```
    lcd.print(millis() / 1000);
```

```
}
```

A.3 Pemrograman *Light Dependent Resistor*

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(10, 9, 8, 7, 6, 5);
int ldr=A0;
int cerah=0,kedip=0;
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);
  lcd.print("Jumlah kedip");
}

void loop() {
  cerah=analogRead(ldr);
  if (cerah >= 20)
  {kedip=kedip+1;}
  lcd.setCursor(0, 1);
  lcd.print(kedip);
  Serial.println(cerah);
  delay(100);
}
```

A.4 Pemrograman *Master*

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <DS3231.h>

DS3231 rtc(SDA, SCL);
Time waktu;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
int sinyal;
int data1, data2;
void setup()
{
  lcd.clear();
  lcd.begin(20, 4);
  lcd.print("PHB GT 1019");
  Serial3.begin(9600);
  Serial.begin(9600);
  rtc.begin();
  data1=4021;
  data2=2025;
}

void loop()
{
  waktu = rtc.getTime();
  int dataJam = waktu.hour;
  int dataMenit = waktu.min;
  int dataDetik = waktu.sec;
  //int dataTanggal = ;

  lcd.setCursor(0, 1);
  //lcd.print("");
  lcd.print(rtc.getDateStr());

  lcd.print("/");
  lcd.print(dataJam);
  lcd.print(':');
  lcd.print(dataMenit);
  lcd.print(':');
```

```

lcd.println(dataDetik);
lcd.setCursor(0, 2);
lcd.print(" Plg-A");
lcd.print(" ");
lcd.print(data1);
lcd.print(" ");
lcd.print("KWh");
lcd.setCursor(0, 3);
lcd.print(" Plg-B");
lcd.print(" ");
lcd.print(data2);
lcd.print(" ");
lcd.print("KWh");
if (Serial3.available()>0){
  sinyal=Serial3.read();
  if(sinyal==97 ){
    data1=data1+1;
  }
  if(sinyal==98){
    data2=data2+1;
  }
}
delay(1);
}

```

A.5 Pemrograman LCD 20X4

```
#include <LiquidCrystal.h>
#include "RTClib.h"
v0,v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15,v16,v17,v18,v
19,v20;
File myFile; //Pendeclarasian File
RTC_DS1307 rtc; //Pendeclarasian RTC 1307

char waktu[20]; //Pendeclarasian Waktu
char Date[20]; //Pendeclarasian Tanggal
char daysOfTheWeek[7][12] = {"Minggu", "Senin", "Selasa", "Rabu",
"Kamis", "Jumat", "Sabtu"};
int ldr=A0;
int ldr_n=0;
void setup () {
  Serial.begin(9600);
  if (! rtc.begin()) { Serial.println("Couldn't find RTC");while(1);}
  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!"); } }
void loop () {
  DateTime now = rtc.now();
  sprintf(waktu, "%02d:%02d:%02d", now.hour(), now.minute(),
now.second());
  sprintf(Date, "%02d/%02d/%02d", now.day(), now.month(),
now.year());
  ldr_n=analogRead(ldr);
  Serial.print(ldr_n); // print ke serial monitor
  Serial.print("\t");
  Serial.println(waktu);
  Serial.print("\t");
  Serial.println(Date);
  myFile = SD.open("Coba3.txt", FILE_WRITE); //membuka file dan
menulis isi filenya
  if (myFile) {
```



```
myFile.print(ldr_n); // print ke file yang ada di micro SD
myFile.print(",");
myFile.println(waktu);
myFile.print(",");
myFile.println(Date);
myFile.close(); // tutup file
}
// jika file tidak dapat dibuka, print error.
else {
    Serial.println("error opening test.txt");
}
delay(1000);
}
```

A.6 Pemrograman Memori Eksternal

```
#include <Wire.h>
```

```
#include "RTCLib.h"
```

```
#include <SD.h>
```

```
File myFile; //Pendeclarasian File
```

```
RTC_DS1307 rtc; //Pendeclarasian RTC 1307
```

```
char Time[20]; //Pendeclarasian Waktu
```

```
char Date[20]; //Pendeclarasian Tanggal
```

```
char daysOfTheWeek[7][12] = {"Sunday", "Monday", "Tuesday",  
"Wednesday", "Thursday", "Friday", "Saturday"};
```

```
int pinCS = 53; // Gunakan pin 10 pada Arduino Uno atau pin 53 pada  
Arduino Mega
```

```
void setup () {
```

```
    Serial.begin(9600);
```

```
    pinMode(pinCS, OUTPUT);
```

```
    // Inisialisasi SD Card
```

```
    if (SD.begin())
```

```
    {
```

```
        Serial.println("SD card is ready to use.");
```

```
    } else
```

```
    {
```

```
        Serial.println("SD card initialization failed");
```

```
        return;
```

```
    }
```

```
    // Inisialisasi RTC
```

```
    if (!rtc.begin()) {
```

```
        Serial.println("Couldn't find RTC");
```

```
        while (1);
```

```
    }
```

```

if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    // Baris berikut menyetel RTC ke tanggal & waktu sketsa ini disusun
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
    // Baris ini menetapkan RTC dengan tanggal & waktu yang jelas,
    misalnya
    // January 21, 2014 pada 3am kamu dapat memanggilnya:
    // rtc.adjust(DateTime(2014, 1, 21, 3, 0, 0));
}
}

void loop () {
    DateTime now = rtc.now();
    sprintf(Time, "%02d:%02d:%02d", now.hour(), now.minute(),
now.second());
    sprintf(Date, "%02d/%02d/%02d", now.day(), now.month(),
now.year());
    Serial.print(Time); // print ke serial monitor
    Serial.print("\t");
    Serial.println(Date);

    myFile = SD.open("Coba11.txt", FILE_WRITE); //membuka file dan
    menulis isi filenya
    if (myFile) {
        myFile.print(Time); // print ke file yang ada di micro SD
        myFile.print(",");
        myFile.println(Date);
        myFile.close(); // tutup file
    }
    // jika file tidak dapat dibuka, print error.
    else {
        Serial.println("error opening test.txt"); }
    delay(1000);}

```

A.7 Pemrograman *Real Time Clock*

```
#include <DS3231.h>
```

```
DS3231 rtc(SDA, SCL);
```

```
Time waktu;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  rtc.begin();
```

```
}
```

```
void loop()
```

```
{
```

```
  //tampilkan hari
```

```
  Serial.println(rtc.getDOWStr(FORMAT_LONG));
```

```
  //ambil waktu
```

```
  waktu = rtc.getTime();
```

```
  int dataJam = waktu.hour;
```

```
  int dataMenit = waktu.min;
```

```
  int dataDetik = waktu.sec;
```

```
  Serial.print(dataJam);
```

```
  Serial.print(':');
```

```
  Serial.print(dataMenit);
```

```
  Serial.print(':');
```

```
  Serial.println(dataDetik);
```

```
  delay(1000);
```

```
}
```

LAMPIRAN B

B.1 Datasheet Real Time Clock

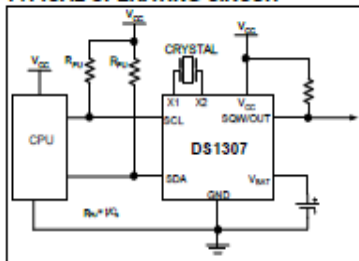


DS1307 64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

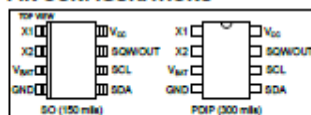
TYPICAL OPERATING CIRCUIT



BENEFITS AND FEATURES

- Completely Manages All Timekeeping Functions
 - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year with Leap-Year Compensation Valid Up to 2100
 - 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
 - Programmable Square-Wave Output Signal
- Simple Serial Port Interfaces to Most Microcontrollers
 - I²C Serial Interface
- Low Power Operation Extends Battery Backup Run Time
 - Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
 - Automatic Power-Fail Detect and Switch Circuitry
- 8-Pin DIP and 8-Pin SO Minimizes Required Space
- Optional Industrial Temperature Range: -40°C to +85°C Supports Operation in a Wide Range of Applications
- Underwriters Laboratories® (UL) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+Denotes a lead-free/Pb-free compliant package.

*A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device. Underwriters Laboratories, Inc. is a registered certification mark of Underwriters Laboratories, Inc.

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature Range (Noncondensing)	
Commercial	0°C to +70°C
Industrial	-40°C to +85°C
Storage Temperature Range	-55°C to +125°C
Soldering Temperature (DIP, leads)	+260°C for 10 seconds
Soldering Temperature (surface mount)	Refer to the J-STD-020 Specification.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS

(T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V _{CC}		4.5	5.0	5.5	V
Logic 1 Input	V _{HI}		2.2		V _{CC} + 0.3	V
Logic 0 Input	V _{LI}		-0.3		+0.8	V
V _{BAT} Battery Voltage	V _{BAT}		2.0	3	3.5	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Leakage (SCL)	I _{LI}		-1		1	μA
I/O Leakage (SDA, SQW/OUT)	I _{LO}		-1		1	μA
Logic 0 Output (I _{OL} = 5mA)	V _{OL}				0.4	V
Active Supply Current (f _{SCL} = 100kHz)	I _{CCA}				1.5	mA
Standby Current	I _{CCS}	(Note 3)			200	μA
V _{BAT} Leakage Current	I _{BATLKG}			5	50	nA
Power-Fail Voltage (V _{BAT} = 3.0V)	V _{PF}		1.216 x V _{BAT}	1.25 x V _{BAT}	1.284 x V _{BAT}	V

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
V _{BAT} Current (OSC ON); SQW/OUT OFF	I _{BAT1}			300	500	nA
V _{BAT} Current (OSC ON); SQW/OUT ON (32kHz)	I _{BAT2}			480	800	nA
V _{BAT} Data-Retention Current (Oscillator Off)	I _{BATDR}			10	100	nA

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
SCL Clock Frequency	f _{SCL}		0		100	kHz
Bus Free Time Between a STOP and START Condition	t _{SUF}		4.7			μs
Hold Time (Repeated) START Condition	t _{HD,STA}	(Note 4)	4.0			μs
LOW Period of SCL Clock	t _{LOW}		4.7			μs
HIGH Period of SCL Clock	t _{HIGH}		4.0			μs
Setup Time for a Repeated START Condition	t _{SU,STA}		4.7			μs
Data Hold Time	t _{HD,DAT}		0			μs
Data Setup Time	t _{SU,DAT}	(Notes 5, 6)	250			ns
Rise Time of Both SDA and SCL Signals	t _r				1000	ns
Fall Time of Both SDA and SCL Signals	t _f				300	ns
Setup Time for STOP Condition	t _{SU,STO}		4.7			μs

CAPACITANCE(T_A = +25°C)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Pin Capacitance (SDA, SCL)	C _{IO}				10	pF
Capacitance Load for Each Bus Line	C _B	(Note 7)			400	pF

Note 1: All voltages are referenced to ground.**Note 2:** Limits at -40°C are guaranteed by design and are not production tested.**Note 3:** I_{CC3} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 6:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V_{SH(MIN)} of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 8:** The maximum t_{HD(DAT)} only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.**Note 7:** C_B—total capacitance of one bus line in pF.

TIMING DIAGRAM

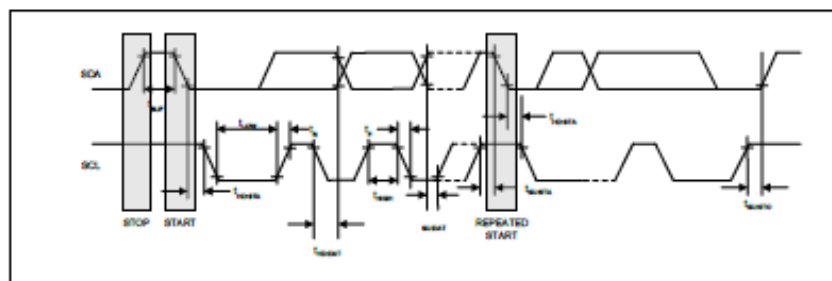
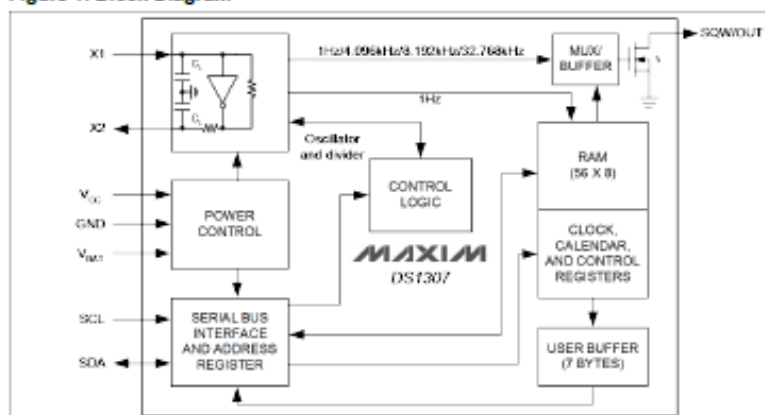
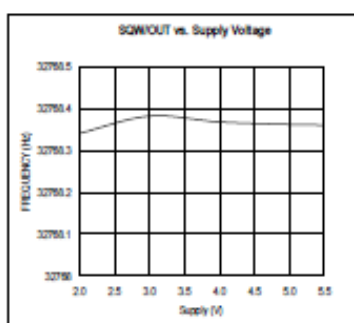
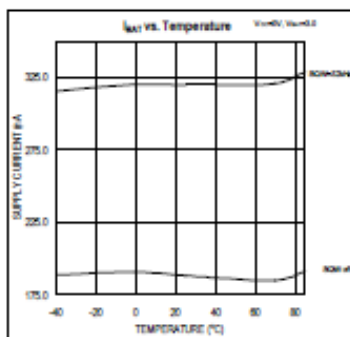
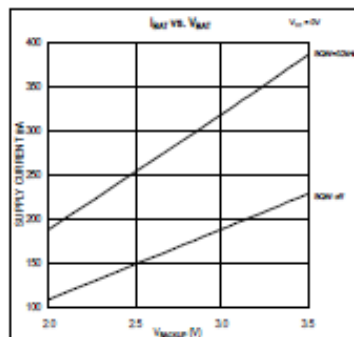
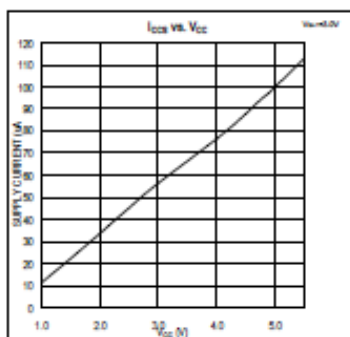


Figure 1. Block Diagram



TYPICAL OPERATING CHARACTERISTICS(V_{CC} = 5.0V, T_A = +25°C, unless otherwise noted.)

PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (C_L) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1.
2	X2	Note: For more information on crystal selection and crystal layout considerations, refer to Application Note 58: Crystal Considerations with Dallas Real-Time Clocks.
3	V _{BAT}	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V _{BAT} pin may prevent proper operation. If a backup supply is not required, V _{BAT} must be grounded. The nominal power-fail trip point (V_{PFR}) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V _{BAT} nominal. A lithium battery with 48mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C. UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: www.maxim-ic.com/ga/info/ul/ .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I ² C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} .
6	SCL	Serial Clock Input. SCL is the clock input for the I ² C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} .
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V _{CC} or V _{BAT} applied. The pullup voltage can be up to 5.5V regardless of the voltage on V _{CC} . If not used, this pin can be left floating.
8	V _{CC}	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V _{CC} is below V _{TR} , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I²C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V_{CC} falls below 1.25 x V_{BAT}, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V_{CC} falls below V_{BAT}, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V_{CC} when V_{CC} is greater than V_{BAT} + 0.2V and recognizes inputs when V_{CC} is greater than 1.25 x V_{BAT}. The block diagram in Figure 1 shows the main elements of the serial RTC.

OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1 shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

CLOCK ACCURACY

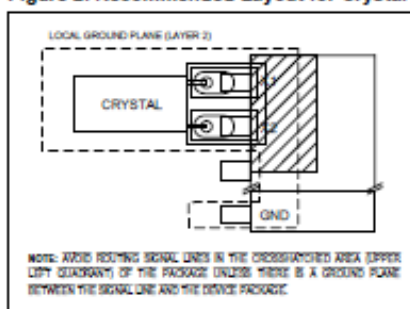
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to Application Note 58: Crystal Considerations with Dallas Real-Time Clocks for detailed information.

Table 1. Crystal Specifications*

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	f_0		32.768		kHz
Series Resistance	ESR			45	k Ω
Load Capacitance	C_L		12.5		pF

*The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to Application Note 58: Crystal Considerations for Dallas Real-Time Clocks for additional specifications.

Figure 2. Recommended Layout for Crystal



RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on.) Illogical time and date entries result in undefined operation. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled. On first application of power to the device the time and date registers are typically reset to 01/01/00 01:00:00 (MM/DD/YY DOW HH:MM:SS). The CH bit in the seconds register will be set to a 1. The clock can be halted whenever the timekeeping functions are not required, which minimizes current (I_{DAVDR}).

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12-hour or 24-hour mode-select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20 to 23 hours). The hours value must be re-entered whenever the 12/24-hour mode bit is changed.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any I²C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I²C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

Table 2. Timekeeper Registers

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours				Hours	1–12 +AM/PM 00–23
		24	PM/AM							
03h	0	0	0	0	0	DAY			Day	01–07
04h	0	0	10 Date		Date				Date	01–31
05h	0	0	0	10 Month	Month				Month	01–12
06h	10 Year		Year						Year	00–99
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh									RAM 56 x 8	00h–FFh

0 = Always reads back as 0.

CONTROL REGISTER

The DS1307 control register is used to control the operation of the SQW/OUT pin.

BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
OUT	0	0	SQWE	0	0	RS1	RS0

Bit 7: Output Control (OUT). This bit controls the output level of the SQW/OUT pin when the square-wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0. On initial application of power to the device, this bit is typically set to a 0.

Bit 4: Square-Wave Enable (SQWE). This bit, when set to logic 1, enables the oscillator output. The frequency of the square-wave output depends upon the value of the RS0 and RS1 bits. With the square-wave output set to 1Hz, the clock registers update on the falling edge of the square wave. On initial application of power to the device, this bit is typically set to a 0.

Bits 1 and 0: Rate Select (RS[1:0]). These bits control the frequency of the square-wave output when the square-wave output has been enabled. The following table lists the square-wave frequencies that can be selected with the RS bits. On initial application of power to the device, these bits are typically set to a 1.

RS1	RS0	SQW/OUT OUTPUT	SQWE	OUT
0	0	1Hz	1	X
0	1	4.095kHz	1	X
1	0	8.192kHz	1	X
1	1	32.768kHz	1	X
X	X	0	0	0
X	X	1	0	1

I²C DATA BUS

The DS1307 supports the I²C protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are referred to as slaves. The bus must be controlled by a master device that generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the I²C bus.

Figures 3, 4, and 5 detail how data is transferred on the I²C bus.

- Data transfer can be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

START data transfer: A change in the state of the data line, from HIGH to LOW, while the clock is HIGH, defines a START condition.

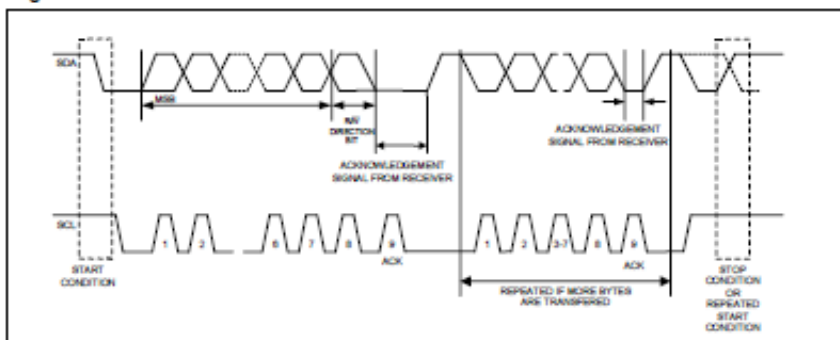
STOP data transfer: A change in the state of the data line, from LOW to HIGH, while the clock line is HIGH, defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the I²C bus specifications a standard mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the standard mode (100kHz) only.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. A master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line HIGH to enable the master to generate the STOP condition.

Figure 3. Data Transfer on I²C Serial Bus

Depending upon the state of the R/w bit, two types of data transfer are possible:

1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a "not acknowledge" is returned.

The master device generates all the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred with the most significant bit (MSB) first.

The DS1307 can operate in the following two modes:

1. **Slave Receiver Mode (Write Mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Hardware performs address recognition after reception of the slave address and direction bit (see Figure 4). The slave address byte is the first byte received after the master generates the START condition. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit (R/w), which for a write is 0. After receiving and decoding the slave address byte, the DS1307 outputs an acknowledge on SDA. After the DS1307 acknowledges the slave address + write bit, the master transmits a word address to the DS1307. This sets the register pointer on the DS1307, with the DS1307 acknowledging the transfer. The master can then transmit zero or more bytes of data with the DS1307 acknowledging each byte received. The register pointer automatically increments after each data byte are written. The master will generate a STOP condition to terminate the data write.
2. **Slave Transmitter Mode (Read Mode):** The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. The DS1307 transmits serial data on SDA while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (see Figure 5). The slave address byte is the first byte received after the START condition is generated by the master. The slave address byte contains the 7-bit DS1307 address, which is 1101000, followed by the direction bit (R/w), which is 1 for a read. After receiving and decoding the slave address the DS1307 outputs an acknowledge on SDA. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The register pointer automatically increments after each byte are read. The DS1307 must receive a Not Acknowledge to end a read.

Figure 4. Data Write—Slave Receiver Mode

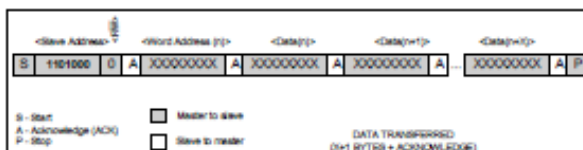
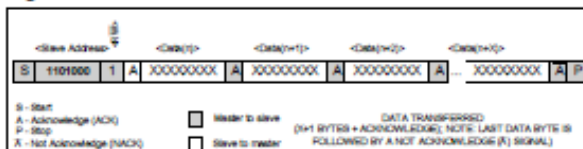


Figure 5. Data Read—Slave Transmitter Mode



REVISION HISTORY

REVISION DATE	DESCRIPTION	PAGES CHANGED
100208	Moved the <i>Typical Operating Circuit</i> and <i>Pin Configurations</i> to first page.	1
	Removed the leaded part numbers from the <i>Ordering Information</i> table.	1
	Added an open-drain transistor to SQW/OUT in the block diagram (Figure 1).	4
	Added the pullup voltage range for SDA, SCL, and SQW/OUT to the <i>Pin Description</i> table and noted that SQW/OUT can be left open if not used.	6
	Added default time and date values on first application of power to the <i>Clock and Calendar</i> section and deleted the note that initial power-on state is not defined.	8
	Added default on initial application of power to bit info in the <i>Control Register</i> section.	9
	Updated the <i>Package Information</i> section to reflect new package outline drawing numbers.	13
3/15	Updated <i>Benefits and Features</i> section	1

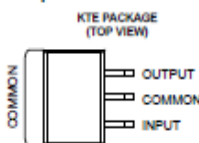
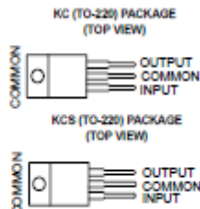
B.2 Datasheet LM7805

LM7800 SERIES POSITIVE-VOLTAGE REGULATORS

SLV095J – MAY 1976 – REVISED MAY 2003

- 3-Terminal Regulators
- Output Current up to 1.5 A
- Internal Thermal-Overload Protection

- High Power-Dissipation Capability
- Internal Short-Circuit Current Limiting
- Output Transistor Safe-Area Compensation



Description/Ordering Information

This series of fixed-voltage integrated-circuit voltage regulators is designed for a wide range of applications. These applications include on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

ORDERING INFORMATION

T _J	V _{O(NOM)} (V)	PACKAGE†	ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 125°C	5	POWER-FLEX (KTE)	Reel of 2000	μA7805CKTER
		TO-220 (KC)	Tube of 50	μA7805CKD
		TO-220, short shoulder (KCS)	Tube of 20	μA7805CKCS
	8	POWER-FLEX (KTE)	Reel of 2000	μA7808CKTER
		TO-220 (KC)	Tube of 50	μA7808CKD
		TO-220, short shoulder (KCS)	Tube of 20	μA7808CKCS
	10	POWER-FLEX (KTE)	Reel of 2000	μA7810CKTER
		TO-220 (KC)	Tube of 50	μA7810CKD
		POWER-FLEX (KTE)	Reel of 2000	μA7812CKTER
	12	TO-220 (KC)	Tube of 50	μA7812CKD
		TO-220, short shoulder (KCS)	Tube of 20	μA7812CKCS
		POWER-FLEX (KTE)	Reel of 2000	μA7815CKTER
	15	TO-220 (KC)	Tube of 50	μA7815CKD
		TO-220, short shoulder (KCS)	Tube of 20	μA7815CKCS
	24	POWER-FLEX (KTE)	Reel of 2000	μA7824CKTER
		TO-220 (KC)	Tube of 50	μA7824CKD

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA INFORMATION IS CONTROLLED BY PUBLICATION DATE.
Products conform to specifications for the period of time indicated by the date of this document.
Standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS
INSTRUMENTS

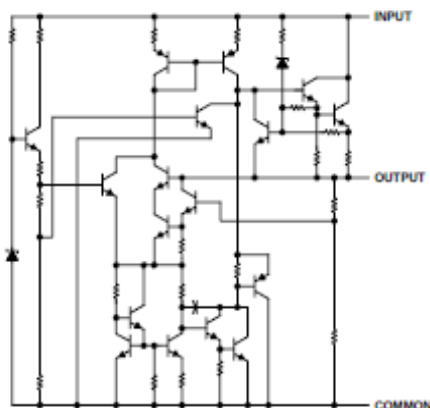
POST OFFICE BOX 655502 • DALLAS, TEXAS 75265

Copyright © 2003, Texas Instruments Incorporated

μA7800 SERIES **POSITIVE-VOLTAGE REGULATORS**

SLV5095J – MAY 1975 – REVISED MAY 2003

schematic



absolute maximum ratings over virtual junction temperature range (unless otherwise noted)[†]

Input voltage, V_i : μA7824C	40 V
All others	35 V
Operating virtual junction temperature, T_J	150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	–65°C to 150°C

[†] Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

package thermal data (see Note 1)

PACKAGE	BOARD	θ_{JC}	θ_{JA}
POWER-FLEX (KTE)	High K, JEDEC 51-5	3°C/W	23°C/W
TO-220 (KCMCS)	High K, JEDEC 51-5	3°C/W	19°C/W

NOTE 1: Maximum power dissipation is a function of $T_J(\max)$, θ_{JA} , and T_A . The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(\max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.

μA7800 SERIES POSITIVE-VOLTAGE REGULATORS

SLV525A – MAY 1976 – REVISED MAY 2003

recommended operating conditions

		MIN	MAX	UNIT
V_I Input voltage	μA7805C	7	25	V
	μA7808C	10.5	25	
	μA7810C	12.5	28	
	μA7812C	14.5	30	
	μA7815C	17.5	30	
	μA7824C	27	38	
I_O Output current			1.5	A
T_J Operating virtual junction temperature	μA7800C series	0	125	°C

electrical characteristics at specified virtual junction temperature, $V_I = 10$ V, $I_O = 500$ mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T _J †	μA7805C			UNIT
			MIN	TYP	MAX	
Output voltage	I _O = 5 mA to 1 A, V _I = 7 V to 20 V, P _D ≤ 15 W	25°C	4.8	5	5.2	V
		0°C to 125°C	4.75		5.25	
Input voltage regulation	V _I = 7 V to 25 V	25°C	3			mV
	V _I = 8 V to 12 V		1			
Ripple rejection	V _I = 8 V to 18 V, f = 120 Hz	0°C to 125°C	62	78		dB
Output voltage regulation	I _O = 5 mA to 1.5 A	25°C	15			mV
	I _O = 250 mA to 750 mA		5			
Output resistance	f = 1 kHz	0°C to 125°C	0.017			Ω
Temperature coefficient of output voltage	I _O = 5 mA	0°C to 125°C	-1.1			mV/°C
Output noise voltage	f = 10 Hz to 100 kHz	25°C	40			μV
Dropout voltage	I _O = 1 A	25°C	2			V
Bias current		25°C	4.2			mA
Bias current change	V _I = 7 V to 25 V	0°C to 125°C	1.3			mA
	I _O = 5 mA to 1 A		0.5			
Short-circuit output current		25°C	750			mA
Peak output current		25°C	2.2			A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

μA7800 SERIES **POSITIVE-VOLTAGE REGULATORS**

SLV026J – MAY 1976 – REVISED MAY 2003

electrical characteristics at specified virtual junction temperature, $V_I = 14\text{ V}$, $I_O = 500\text{ mA}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J †	μA7800C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$, $V_I = 10.5\text{ V to }23\text{ V}$, $P_D \leq 15\text{ W}$	25°C	7.7	8	8.3	V
		0°C to 125°C	7.6		8.4	
Input voltage regulation	$V_I = 10.5\text{ V to }25\text{ V}$	25°C		6	160	mV
	$V_I = 11\text{ V to }17\text{ V}$			2	80	
Ripple rejection	$V_I = 11.5\text{ V to }21.5\text{ V}$, $f = 120\text{ Hz}$	0°C to 125°C	55	72		dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$	25°C		12	160	mV
	$I_O = 250\text{ mA to }750\text{ mA}$			4	80	
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.016		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		-0.8		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		52		μV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.3	8	mA
	$V_I = 10.5\text{ V to }25\text{ V}$	0°C to 125°C			1	
Bias current change	$I_O = 5\text{ mA to }1\text{ A}$				0.5	mA
		25°C		450		
Short-circuit output current		25°C		450		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.

electrical characteristics at specified virtual junction temperature, $V_I = 17\text{ V}$, $I_O = 500\text{ mA}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J †	μA7810C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$, $V_I = 12.5\text{ V to }25\text{ V}$, $P_D \leq 15\text{ W}$	25°C	9.6	10	10.4	V
		0°C to 125°C	9.5	10	10.5	
Input voltage regulation	$V_I = 12.5\text{ V to }28\text{ V}$	25°C		7	200	mV
	$V_I = 14\text{ V to }20\text{ V}$			2	100	
Ripple rejection	$V_I = 13\text{ V to }23\text{ V}$, $f = 120\text{ Hz}$	0°C to 125°C	55	71		dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$	25°C		12	200	mV
	$I_O = 250\text{ mA to }750\text{ mA}$			4	100	
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.018		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		-1		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		70		μV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.3	8	mA
	$V_I = 12.5\text{ V to }28\text{ V}$	0°C to 125°C			1	
Bias current change	$I_O = 5\text{ mA to }1\text{ A}$				0.5	mA
		25°C		400		
Short-circuit output current		25°C		400		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-μF capacitor across the input and a 0.1-μF capacitor across the output.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

µA7800 SERIES POSITIVE-VOLTAGE REGULATORS

SLS0256J – MAY 1976 – REVISED MAY 2003

electrical characteristics at specified virtual junction temperature, $V_I = 19\text{ V}$, $I_O = 500\text{ mA}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J †	µA7812C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$, $V_I = 14.5\text{ V to }27\text{ V}$, $P_D \leq 15\text{ W}$	25°C 0°C to 125°C	11.5 11.4	12	12.5 12.6	V
Input voltage regulation	$V_I = 14.5\text{ V to }30\text{ V}$ $V_I = 16\text{ V to }22\text{ V}$	25°C		10 3	240 120	mV
Ripple rejection	$V_I = 15\text{ V to }25\text{ V}$, $f = 120\text{ Hz}$	0°C to 125°C		55	71	dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$ $I_O = 250\text{ mA to }750\text{ mA}$	25°C		12 4	240 120	mV
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.018		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		-1		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		75		µV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.3	8	mA
Bias current change	$V_I = 14.5\text{ V to }30\text{ V}$ $I_O = 5\text{ mA to }1\text{ A}$	0°C to 125°C			1 0.5	mA
Short-circuit output current		25°C		350		mA
Peak output current		25°C		2.2		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-µF capacitor across the input and a 0.1-µF capacitor across the output.

electrical characteristics at specified virtual junction temperature, $V_I = 23\text{ V}$, $I_O = 500\text{ mA}$ (unless otherwise noted)

PARAMETER	TEST CONDITIONS	T_J †	µA7815C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5\text{ mA to }1\text{ A}$, $V_I = 17.5\text{ V to }30\text{ V}$, $P_D \leq 15\text{ W}$	25°C 0°C to 125°C	14.4 14.25	15	15.6 15.75	V
Input voltage regulation	$V_I = 17.5\text{ V to }30\text{ V}$ $V_I = 20\text{ V to }28\text{ V}$	25°C		11 3	300 150	mV
Ripple rejection	$V_I = 18.5\text{ V to }28.5\text{ V}$, $f = 120\text{ Hz}$	0°C to 125°C		54	70	dB
Output voltage regulation	$I_O = 5\text{ mA to }1.5\text{ A}$ $I_O = 250\text{ mA to }750\text{ mA}$	25°C		12 4	300 150	mV
Output resistance	$f = 1\text{ kHz}$	0°C to 125°C		0.019		Ω
Temperature coefficient of output voltage	$I_O = 5\text{ mA}$	0°C to 125°C		-1		mV/°C
Output noise voltage	$f = 10\text{ Hz to }100\text{ kHz}$	25°C		90		µV
Dropout voltage	$I_O = 1\text{ A}$	25°C		2		V
Bias current		25°C		4.4	8	mA
Bias current change	$V_I = 17.5\text{ V to }30\text{ V}$ $I_O = 5\text{ mA to }1\text{ A}$	0°C to 125°C			1 0.5	mA
Short-circuit output current		25°C		230		mA
Peak output current		25°C		2.1		A

† Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33-µF capacitor across the input and a 0.1-µF capacitor across the output.



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

LM7800 SERIES **POSITIVE-VOLTAGE REGULATORS**

SLV056J – MAY 1976 – REVISED MAY 2003

electrical characteristics at specified virtual junction temperature, $V_I = 33$ V, $I_O = 500$ mA (unless otherwise noted)

PARAMETER	TEST CONDITIONS	$T_J^†$	LM7824C			UNIT
			MIN	TYP	MAX	
Output voltage	$I_O = 5$ mA to 1 A, $V_I = 27$ V to 38 V, $P_D \leq 15$ W	25°C	23	24	25	V
		0°C to 125°C	22.8		25.2	
Input voltage regulation	$V_I = 27$ V to 38 V	25°C		18	480	mV
	$V_I = 30$ V to 36 V			6	240	
Ripple rejection	$V_I = 28$ V to 38 V, $f = 120$ Hz	0°C to 125°C	50	66		dB
Output voltage regulation	$I_O = 5$ mA to 1.5 A	25°C		12	480	mV
	$I_O = 250$ mA to 750 mA			4	240	
Output resistance	$f = 1$ kHz	0°C to 125°C		0.028		Ω
Temperature coefficient of output voltage	$I_O = 5$ mA	0°C to 125°C		-1.5		mV/°C
Output noise voltage	$f = 10$ Hz to 100 kHz	25°C		170		μ V
Dropout voltage	$I_O = 1$ A	25°C		2		V
Bias current		25°C		4.6	8	mA
Bias current change	$V_I = 27$ V to 38 V	0°C to 125°C			1	mA
	$I_O = 5$ mA to 1 A				0.5	
Short-circuit output current		25°C		150		mA
Peak output current		25°C		2.1		A

[†] Pulse-testing techniques maintain the junction temperature as close to the ambient temperature as possible. Thermal effects must be taken into account separately. All characteristics are measured with a 0.33- μ F capacitor across the input and a 0.1- μ F capacitor across the output.

APPLICATION INFORMATION

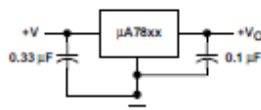


Figure 1. Fixed-Output Regulator

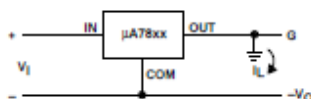
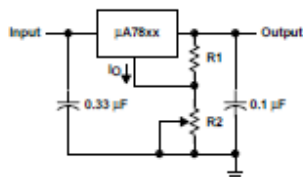


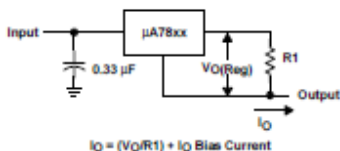
Figure 2. Positive Regulator in Negative Configuration (V_I Must Float)



NOTE A: The following formula is used when V_{xx} is the nominal output voltage (output to common) of the fixed regulator:

$$V_O = V_{xx} + \left(\frac{V_{xx}}{R1} + I_O \right) R2$$

Figure 3. Adjustable-Output Regulator



$$I_O = (V_O/R1) + I_O \text{ Bias Current}$$

Figure 4. Current Regulator

μA7800 SERIES **POSITIVE-VOLTAGE REGULATORS**

SLV506J – MAY 1976 – REVISED MAY 2003

APPLICATION INFORMATION

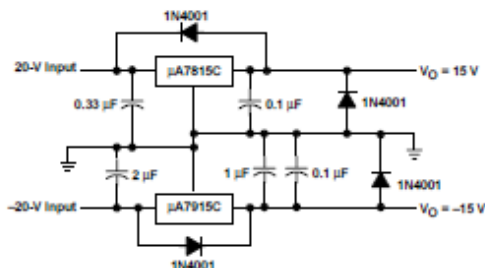


Figure 5. Regulated Dual Supply

operation with a load common to a voltage of opposite polarity

In many cases, a regulator powers a load that is not connected to ground but, instead, is connected to a voltage source of opposite polarity (e.g., operational amplifiers, level-shifting circuits, etc.). In these cases, a clamp diode should be connected to the regulator output as shown in Figure 6. This protects the regulator from output polarity reversals during startup and short-circuit operation.

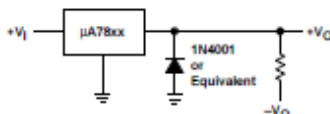


Figure 6. Output Polarity-Reversal-Protection Circuit

reverse-bias protection

Occasionally, the input voltage to the regulator can collapse faster than the output voltage. This can occur, for example, when the input supply is crowbarred during an output overvoltage condition. If the output voltage is greater than approximately 7 V, the emitter-base junction of the series-pass element (internal or external) could break down and be damaged. To prevent this, a diode shunt can be used as shown in Figure 7.

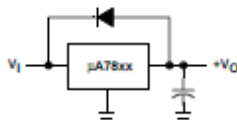


Figure 7. Reverse-Bias-Protection Circuit



**TEXAS
INSTRUMENTS**

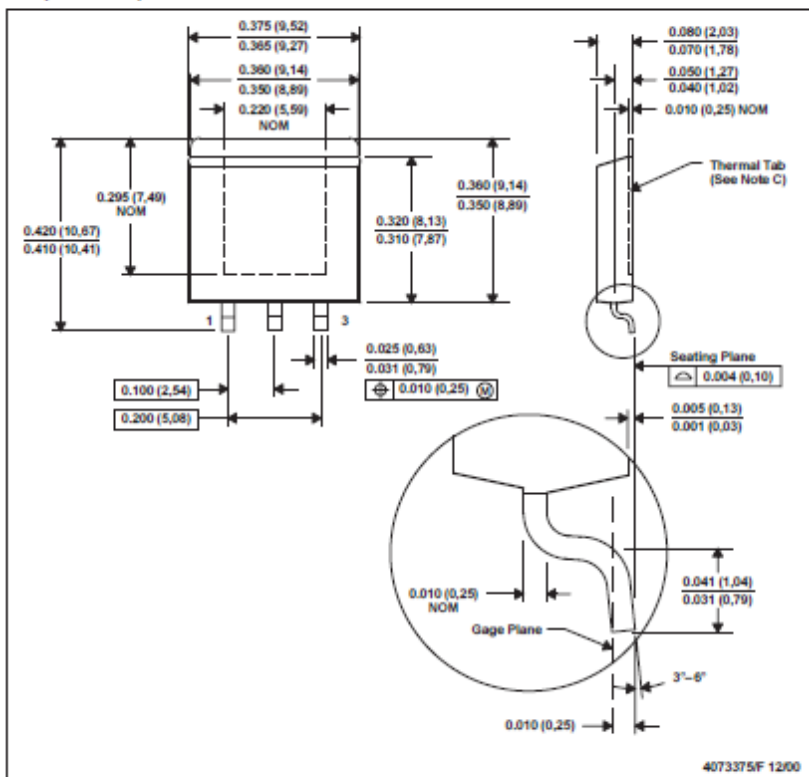
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

MECHANICAL DATA

MPFMD01E - OCTOBER 1994 - REVISED JANUARY 2001

KTE (R-PSFM-G3)

PowerFLEX™ PLASTIC FLANGE-MOUNT



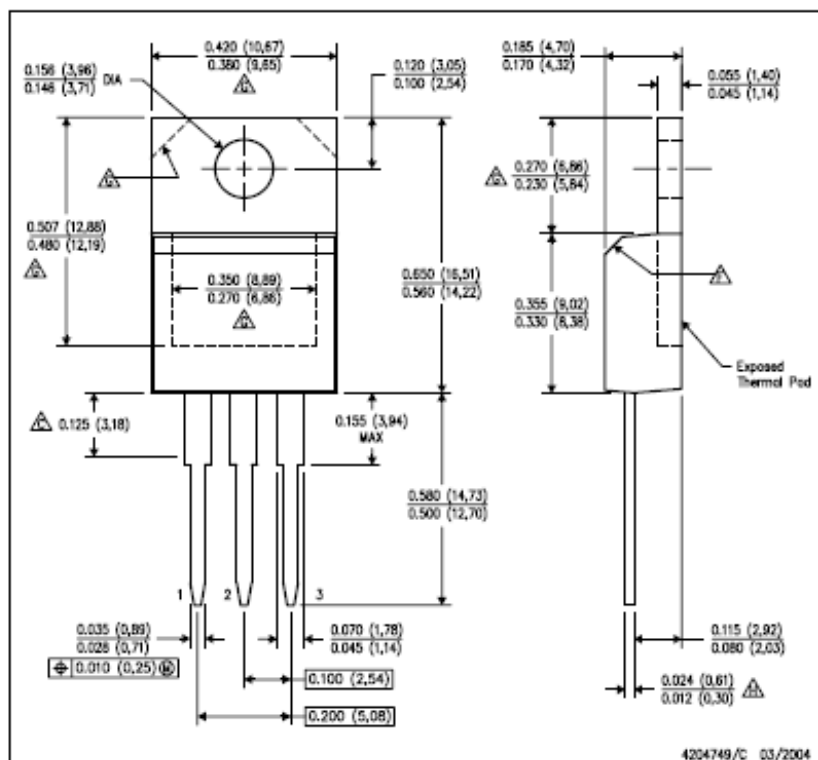
- NOTES:
- All linear dimensions are in inches (millimeters).
 - This drawing is subject to change without notice.
 - The center lead is in electrical contact with the thermal tab.
 - Dimensions do not include mold protrusions, not to exceed 0.006 (0,15).
 - Falls within JEDEC MO-169

PowerFLEX is a trademark of Texas Instruments.

**TEXAS
INSTRUMENTS**
POST OFFICE BOX 955303 • DALLAS, TEXAS 75295

KCS (R-PSFM-T3)

PLASTIC FLANGE-MOUNT PACKAGE

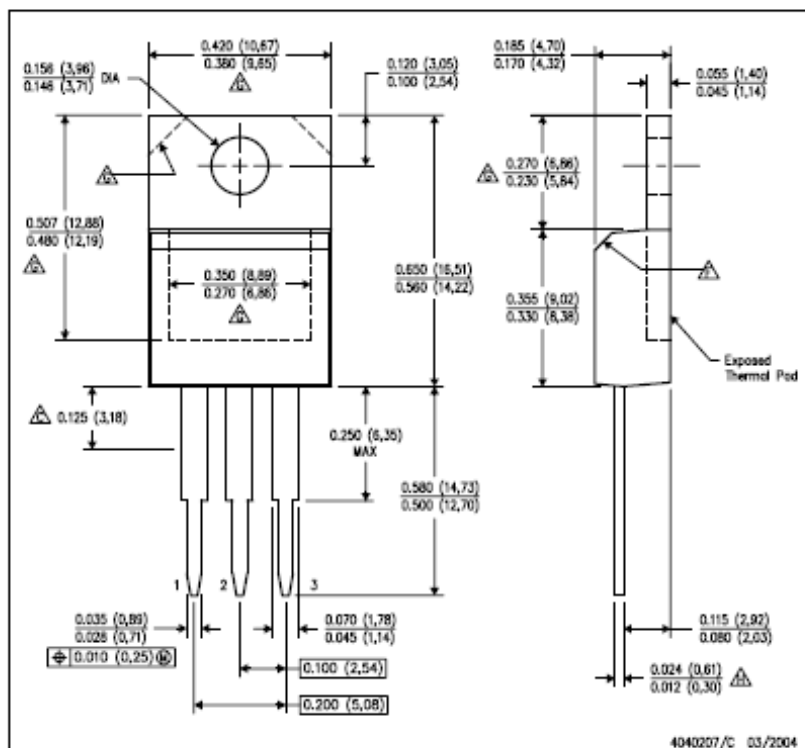


4204749/C 03/2004

- NOTES:
- All linear dimensions are in inches (millimeters).
 - This drawing is subject to change without notice.
 - Lead dimensions are not controlled within this area.
 - All lead dimensions apply before solder dip.
 - The center lead is in electrical contact with the mounting tab.
 - The chamfer is optional.
 - Thermal pad contour optional within these dimensions.
 - Falls within JEDEC TO-220 variation AB, except minimum lead thickness.

KC (R-PSFM-T3)

PLASTIC FLANGE-MOUNT PACKAGE



NOTES:

- All linear dimensions are in inches (millimeters).
- This drawing is subject to change without notice.
- Lead dimensions are not controlled within this area.
- All lead dimensions apply before solder dip.
- The center lead is in electrical contact with the mounting tab.
- The chamfer is optional.
- Thermal pad contour optional within these dimensions.
- Falls within JEDEC TO-220 variation AB, except minimum lead thickness.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2004, Texas Instruments Incorporated

B.3 Datasheet Adruino Uno



[Buy](#) [Download](#) [Getting Started](#) [Learning](#) [Reference](#) [Hardware](#) [FAQ](#)

Arduino Uno



Overview

The Arduino Uno is a microcontroller board based on the ATmega328P ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Uno board has a resistor pulling the 8U2

HWB line to ground, making it easier to put into **DFU mode**.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-uno-reference-design.zip](#)

Schematic: [arduino-uno-schematic.pdf](#)

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- ✦ **VIN**. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- ✦ **5V**. The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- ✦ **3V3**. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- ✦ **GND**. Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

✦ **Serial:** 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

✦ **External Interrupts:** 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.

✦ **PWM:** 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

✦ **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the [SPI library](#).

✦ **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

✦ **TWI:** A4 (SDA) and A5 (SCL). Support TWI communication using the [Wire library](#).

There are a couple of other pins on the board:

✦ **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).

✦ **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and ATmega328 ports](#)⁷.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual COM port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, [on Windows, a .inf file is required](#). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a [Wire library](#) to simplify use of the I2C bus; see the [documentation](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [Chandler files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available. The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

U S B O v e r c u r r e n t P r o t e c t i o n

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

P h y s i c a l C h a r a c t e r i s t i c s

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

 Share |    

B.4 Datasheet Arduino Mega 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical
Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms &
Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

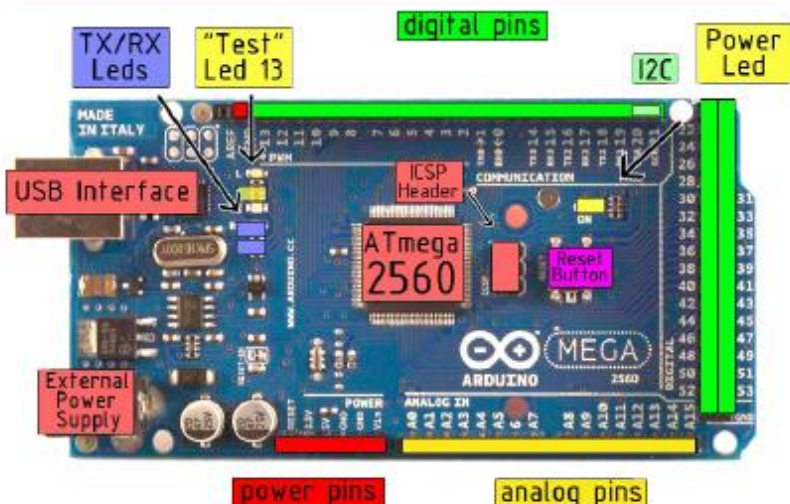


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (Interrupt 0), 3 (Interrupt 1), 18 (Interrupt 5), 19 (Interrupt 4), 20 (Interrupt 3), and 21 (Interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM:** 0 to 13. Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C:** 20 (SDA) and 21 (SCL). Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your skeeth you'll see something very close to the screenshot on the right.

In Tools>Board select MEGA

Now you have to go to
Tools>SerialPort
and select the right serial port, the one arduino is attached to.



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



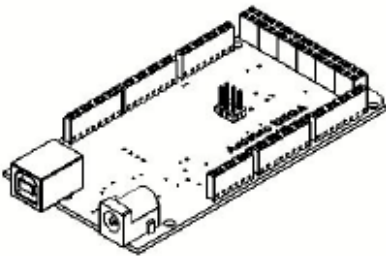
Blinking Led!



radiospares

RADIONICS





Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or misstatement by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, the producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS



RIWAYAT PENULIS



Nama : Biner Rochmatul C.A
TTL : Mojokerto, 14 April 1996
Jenis Kelamin : Perempuan
Agama : Islam
Alamat Asal : RT 03 RW 02 No. 31
Ds.Jatipelem, Kec. Diwek,
Kab. Jombang
Telp/HP : 081252789767
E-mail : *binerrochmatul@gmail.com*

RIWAYAT PENDIDIKAN

- 2002 – 2008 : MIN Kauman Utara Jombang
- 2008 – 2011 : MTsN Denanyar Jombang
- 2011 – 2014 : MAN 3 Malang
- 2014 – Sekarang : Departemen Teknik Elektro Otomasi Bidang Studi Teknik Listrik Institut Teknologi Sepuluh Nopember.

PENGALAMAN KERJA

- Kerja Praktek PT PLN (Persero) Distribusi Jawa Timur Area Surabaya Selatan Rayon Ngagel (Januari 2017 – Maret 2017)

--- Halaman ini sengaja dikosongkan ---

RIWAYAT PENULIS



Nama : M. Fatkhur Rozi
TTL : Jombang, 14 Mei 1994
Jenis Kelamin : Laki-Laki
Agama : Islam
Alamat Asal : RT 03 RW 03 No. 31 Desa
Pulosari Kecamatan Bareng
Kabupaten Jombang
Telp/HP : 085649475688
E-mail : *mfrozik@gmail.com*

RIWAYAT PENDIDIKAN

- 2001 – 2007 : SDN Pulosari I Bareng
- 2007 – 2010 : SMPN 2 Mojowarno
- 2010 – 2013 : SMAN 2 Jombang
- 2014 – Sekarang : Departemen Teknik Elektro Otomasi Bidang Studi Teknik Listrik Institut Teknologi Sepuluh Nopember.

PENGALAMAN KERJA

- Kerja Praktek PT PLN (Persero) Distribusi Jawa Timur Area Surabaya Selatan Rayon Ngagel (Januari 2017 – Maret 2017)

--- *Halaman ini sengaja dikosongkan* ---